Étude des techniques d'injection de fautes par violation de contraintes temporelles permettant la cryptanalyse physique de circuits sécurisés

Loic Zussa

NNT : 2014 EMSE 0757

THÈSE

Présentée par

Loïc ZUSSA

pour obtenir le grade de

Docteur de l'École Nationale Supérieure
des Mines de Saint-Étienne

Spécialité : MICROÉLECTRONIQUE

Étude des techniques d'injection de fautes par violation de contraintes temporelles permettant la cryptanalyse physique de circuits sécurisés

soutenue à Gardanne le 10 Octobre 2014

Jury :

Président :	Jean-Luc AUTRAN	-	Aix-Marseille Université, Marseille
Rapporteurs :	Jean-Luc DANGER	-	Telecom ParisTech, Paris
	Viktor FISCHER	-	Université Jean Monnet, Saint Etienne
Examinateurs :	Lorena ANGHEL	-	Grenoble INP, Grenoble
	Giorgio DI NATALE	-	LIRMM, Montpellier
Directeur :	Assia TRIA	-	CEA-TECH, Gardanne
Encadrants :	Jean-Max DUTERTRE	-	MINES Saint-Étienne, Gardanne
	Jessy CLÉDIÈRE	-	CEA-LETI, Grenoble
Invité :	Bruno ROBISSON	-	CEA-TECH, Gardanne

Spécialités doctorales	Responsables :	Spécialités doctorales	Responsables
SCIENCES ET GENIE DES MATERIAUX	K. Wolski Directeur de recherche	MATHEMATIQUES APPLIQUEES	O. Roustant, Maître-assistant
MECANIQUE ET INGENIERIE	S. Drapier, professeur	INFORMATIQUE	O. Boissier, Professeur
GENIE DES PROCEDES	F. Gruy, Maître de recherche	IMAGE, VISION, SIGNAL	JC. Pinoli, Professeur
SCIENCES DE LA TERRE	B. Guy, Directeur de recherche	GENIE INDUSTRIEL	A. Dolgui, Professeur
SCIENCES ET GENIE DE L'ENVIRONNEMENT	D. Graillot, Directeur de recherche	MICROELECTRONIQUE	S. Dauzere Peres, Professeur

EMSE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

ABSI	Nabil	CR		CMP
AVRIL	Stéphane	PR2	Mécanique et ingénierie	CIS
BALBO	Flavien	PR2		FAYOL
BASSEREAU	Jean-François	PR		SMS
BATTON-HUBERT	Mireille	PR2	Sciences et génie de l'environnement	FAYOL
BERGER DOUCE	Sandrine	PR2		FAYOL
BERNACHE-ASSOLLANT	Didier	PR0	Génie des Procédés	CIS
BIGOT	Jean Pierre	MR(DR2)	Génie des Procédés	SPIN
BILAL	Essaid	DR	Sciences de la Terre	SPIN
BOISSIER	Olivier	PR1	Informatique	FAYOL
BORBELY	Andras	MR(DR2)	Sciences et génie des matériaux	SMS
BOUCHER	Xavier	PR2	Génie Industriel	FAYOL
BRODHAG	Christian	DR	Sciences et génie de l'environnement	FAYOL
BRUCHON	Julien	MA(MDC)	Mécanique et ingénierie	SMS
BURLAT	Patrick	PR2	Génie Industriel	FAYOL
COURNIL	Michel	PR0	Génie des Procédés	DIR
DARRIEULAT	Michel	IGM	Sciences et génie des matériaux	SMS
DAUZERE-PERES	Stéphane	PR1	Génie Industriel	CMP
DEBAYLE	Johan	CR	Image Vision Signal	CIS
DELAFOSSE	David	PR1	Sciences et génie des matériaux	SMS
DESRAYAUD	Christophe	PR2	Mécanique et ingénierie	SMS
DOLGUI	Alexandre	PR0	Génie Industriel	FAYOL
DRAPIER	Sylvain	PR1	Mécanique et ingénierie	SMS
FEILLET	Dominique	PR2	Génie Industriel	CMP
FEVOTTE	Gilles	PR1	Génie des Procédés	SPIN
FRACZKIEWICZ	Anna	DR	Sciences et génie des matériaux	SMS
GARCIA	Daniel	MR(DR2)	Génie des Procédés	SPIN
GERINGER	Jean	MA(MDC)	Sciences et génie des matériaux	CIS
GOEURIOT	Dominique	DR	Sciences et génie des matériaux	SMS
GRAILLOT	Didier	DR	Sciences et génie de l'environnement	SPIN
GROSSEAU	Philippe	DR	Génie des Procédés	SPIN
GRUY	Frédéric	PR1	Génie des Procédés	SPIN
GUY	Bernard	DR	Sciences de la Terre	SPIN
HAN	Woo-Suck	CR	Mécanique et ingénierie	SMS
HERRI	Jean Michel	PR1	Génie des Procédés	SPIN
KERMOUCHE	Guillaume	PR2	Mécanique et Ingénierie	SMS
KLOCKER	Helmut	DR	Sciences et génie des matériaux	SMS
LAFOREST	Valérie	MR(DR2)	Sciences et génie de l'environnement	FAYOL
LERICHE	Rodolphe	CR	Mécanique et ingénierie	FAYOL
LI	Jean-Michel		Microélectronique	CMP
MALLIARAS	Georges	PR1	Microélectronique	CMP
MOLIMARD	Jérôme	PR2	Mécanique et ingénierie	CIS
MONTHEILLET	Frank	DR	Sciences et génie des matériaux	SMS
MOUTTE	Jacques	CR	Génie des Procédés	SPIN
NEUBERT	Gilles			FAYOL
NIKOLOVSKI	Jean-Pierre			CMP
NORTIER	Patrice	PR1		SPIN
PIJOLAT	Christophe	PR0	Génie des Procédés	SPIN
PIJOLAT	Michèle	PR1	Génie des Procédés	SPIN
PINOLI	Jean Charles	PR0	Image Vision Signal	CIS
POURCHEZ	Jérémy	CR	Génie des Procédés	CIS
ROBISSON	Bruno			CMP
ROUSSY	Agnès	MA(MDC)		CMP
ROUSTANT	Olivier	MA(MDC)		FAYOL
ROUX	Christian	PR		CIS
STOLARZ	Jacques	CR	Sciences et génie des matériaux	SMS
TRIA	Assia	Ingénieur de recherche	Microélectronique	CMP
VALDIVIESO	François	MA(MDC)	Sciences et génie des matériaux	SMS
VIRICELLE	Jean Paul	MR(DR2)	Génie des Procédés	SPIN
WOLSKI	Krzystof	DR	Sciences et génie des matériaux	SMS
XIE	Xiaolan	PR1	Génie industriel	CIS
YUGMA	Gallian	CR	Génie industriel	CMP

ENISE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

BERGHEAU	Jean-Michel	PU	Mécanique et Ingénierie	ENISE
BERTRAND	Philippe	MCF	Génie des procédés	ENISE
DUBUJET	Philippe	PU	Mécanique et Ingénierie	ENISE
FEULVARCH	Eric	MCF	Mécanique et Ingénierie	ENISE
FORTUNIER	Roland	PR	Sciences et Génie des matériaux	ENISE
GUSSAROV	Andrey	Enseignant contractuel	Génie des procédés	ENISE
HAMDI	Hédi	MCF	Mécanique et Ingénierie	ENISE
LYONNET	Patrick	PU	Mécanique et Ingénierie	ENISE
RECH	Joël	PU	Mécanique et Ingénierie	ENISE
SMUROV	Igor	PU	Mécanique et Ingénierie	ENISE
TOSCANO	Rosario	PU	Mécanique et Ingénierie	ENISE
ZAHOUANI	Hassan	PU	Mécanique et Ingénierie	ENISE

Mise à jour : 28/03/2014

Remerciements

Ce travail de thèse a été réalisé au Centre Microélectronique de Provence (CMP) à Gardanne (France), au sein du laboratoire Systèmes et Architectures Sécurisés (SAS). Cette équipe mixte CEA-Tech et MINES Saint-Étienne est dirigée par Monsieur Bruno Robisson.

A l'issue de cette thèse préparée entre octobre 2011 et septembre 2014, je tiens à adresser mes remerciements à :

Madame Assia Tria qui, après avoir été ma professeur de cryptographie en école d'ingénieur, a dirigé mes travaux de thèse. Je la remercie particulièrement pour son aide attentive lors de la relecture de ce manuscrit.

Monsieur Bruno Robisson pour les échanges enrichissants que nous avons eus et pour avoir dirigé une partie de mes travaux.

Monsieur Jean-Max Dutertre pour avoir encadré mes travaux avec beaucoup de disponibilité et de pédagogie tout au long de ces trois années. Mes remerciements à son égard sont doubles puisqu'il a aussi été mon professeur d'électronique et qu'il m'a proposé à l'issue de ma formation d'ingénieur de poursuivre en thèse sur le sujet dont ce manuscrit est le résultat.

Monsieur Jessy Clédière pour avoir accepté de co-encadrer mes travaux de thèse afin d'apporter un regard extérieur et critique vis-à-vis de mon travail.

Professeur Jean-Luc Danger et Professeur Viktor Fischer pour avoir accepté la charge d'être les rapporteurs de ce manuscrit et pour l'intérêt qu'ils ont porté à mes travaux.

Professeur Jean-Luc Autran de l'honneur qu'il m'a fait d'avoir accepté d'être examinateur de mes travaux pour la seconde fois (la première étant lors de ma soutenance de master).

Professeur Lorena Anghel qui a accepté d'être examinatrice et qui a pu, malgré les coups du sort, participer pleinement via visoconférence à ma soutenance.

Monsieur Giorgio Di Natale que j'avais eu le plaisir de rencontrer à plusieurs reprises en conférence pour sa présence en tant qu'examinateur lors de ma soutenance.

Durant la préparation de ma thèse, j'ai eu l'occasion de côtoyer de nombreuses personnes dont les personnalités uniques ont fait de ces trois années une expérience professionnelle mais aussi une expérience personnelle passionnante et enrichissante dont je garde un excellent souvenir.

Je salue Jean-Baptiste Rigaud et Bernard Dhalluin qui ont participé significativement à mon orientation professionnelle : des salles de cours jusqu'à la C201.

J'aimerais aussi remercier toute l'équipe qui m'a offert un cadre de travail agréable que je quitte avec, déjà, un peu de nostalgie. Je remercie notamment Anne-Lise Ribotta, Loïc Lauro, Olivier Vallier et Amine Dehbaoui pour leur support technique. Je remercie aussi chaudement Véronique Villaréal, Michelle Gillet et Barbara Bruno pour leur disponibilité constante.

J'ai une pensée amicale pour Ingrid Exurville, Nicolas Moro, Thomas Sarno, Marc Ferro, Patrick Haddad, Marc Lacruche, Hélène Le Bouder, Ronan Lashermes, Cyril Roscian, Florian Praden, Alexandre Sarafianos, Stéphan De Castro, Franck Courbon, Kamil Gomina, Manuelle Bongo, Clément Talagrand, Nicolas Borrel, Clément Champeix, Thierry Vaschalde, Driss Aboulkassimi, Amir-Pasha Mirbaha, Loïc Lauro, Guillaume Reymond et Sebastien Tiran qui ont partagé avec moi des formations, des conférences, des pauses cafés, des restaurants d'anniversaire ou l'espace thésards.

Enfin, j'adresse ma gratitude toute particulière à ma famille qui m'a donné l'éducation et le soutien permanent qui m'ont permit de suivre la voie que je souhaitais, à Nicolas qui a été mon co-thésard mais aussi mon colocataire pendant ces trois années de thèse et à Ingrid qui me soutient au quotidien, m'encourage et me supporte dans les bons et les moins bons moments de ma vie.

"Pour connaître la valeur d'une heure, interroge l'amoureux qui attend son rendez-vous. Pour connaître la valeur d'une minute, interroge l'homme pressé qui vient de rater son bus. Pour connaître la valeur d'une seconde, interroge celui qui a perdu un être cher dans un accident de voiture. Pour connaître la valeur d'un millième de seconde interroge le médaillé d'une finale olympique."

Bernard Werber

Table des matières

Publications

[Zussa 2012] Loïc Zussa, Jean-Max Dutertre, Jessy Clédière, Bruno Robisson et Assia Tria. *Investigation of timing constraints violation as a fault injection means.* Design of Circuits and Integrated Systems (DCIS), 2012. (Non cité.)

[Zussa 2013] Loïc Zussa, Jean-Max Dutertre, Jessy Clediere et Assia Tria. *Power supply glitch induced faults on FPGA : An in-depth analysis of the injection mechanism.* On-Line Testing Symposium (IOLTS), 2013. (Non cité.)

[Zussa 2014a] Loïc Zussa, Amine Dehbaoui, Karim Tobich, Jean-max Dutertre, Philippe Maurine, Ludovic Guillaume-sage, Jessy Clediere et Assia Tria. *Efficiency of a Glitch Detector against Electromagnetic Fault Injection.* Design, Automation & Test in Europe (DATE), 2014. (Non cité.)

[Zussa 2014b] Loïc Zussa, Jean-Max Dutertre, Jessy Clédière et Bruno Robisson. *Analysis of the fault injection mechanism related to negative and positive power supply glitches using an on-chip voltmeter.* Hardware-Oriented Security and Trust (HOST), 2014. (Non cité.)

Glossaire

AES : Advanced Encryption Standard
CPA : Correlation Power Analysis - *Analyse de la consommation par corrélation*
DBA : Differential Behavior Analysis - *Analyse différentielle du comportement*
DFF : D Flip flop - *Bascule D*
DPA : Differential Power Analysis - *Analyse différentielle de la consommation*
DPL : Dual rail Precharged Logic - *Logique duale préchargée*
DVCT : Détecteur de Violation de Contrainte Temporelle
FA : Fault Attack - *Attaque en faute*
FPGA : Field Programmable Gate Array - *Matrice de portes programmables*
FSA : Fault Sensitivity Analysis - *Analyse de la sensibilité à l'injection de faute*
GE : Guessing Entropy - *Indice d'incertitude*
HW : Hamming Weight - *Poids de Hamming*
MIA : Mutual Information Analysis - *Analyse de l'information mutuelle*
NIST : National Institute of Standards and Technology
PLL : Phase Locked Loop - *Boucle à verrouillage de phase*
SCA : Side Channel Analysis - *Analyse par canal auxiliaire*
SHW : Signed Hamming Weight - *Poids de Hamming signé*
TA : Template Attack - *Attaque par dictionnaire*
TDC : Time to Digital Convertor - *Convertisseur temporel vers numérique*

Introduction générale

Aujourd'hui, la plupart des composants électroniques communicants ou d'identification (smart cards, téléphones mobiles, télévision à péage, passeport) embarquent des systèmes cryptographiques. Ces systèmes permettent à leurs utilisateurs de bénéficier de services de confidentialité, d'authenticité et d'assurer l'intégrité de leurs données personnelles. Le dispositif le plus répandu aujourd'hui implémentant des algorithmes cryptographiques est la carte à puce. Ces circuits manipulant des données sensibles peuvent être la cible d'attaquants tentant d'extraire ces informations confidentielles. En effet, si un algorithme cryptographique est réputé mathématiquement robuste, son implémentation matérielle peut être vulnérable aux attaques physiques. Il existe aujourd'hui un très large panel d'attaques visant l'intégrité de ces systèmes. La caractérisation du niveau de sécurité de tels circuits est une tâche qui peut s'avérer particulièrement difficile.

Il existe trois différentes techniques pour mener ce genre d'attaques physiques :

- La première consiste à obtenir des informations sur la conception du circuit par analyse directe de son implémentation matérielle.
- La seconde consiste à observer les variations comportementales du circuit (consommation électrique, radiations électromagnétiques, temps de réponse, etc.) qui peuvent dépendre dans une certaine mesure des données sensibles manipulées.
- La troisième consiste à perturber le fonctionnement du circuit afin d'obtenir une erreur qui peut, dans certains cas, permettre de retrouver les informations secrètes.

Aussi, l'étude des mécanismes d'injection de fautes pouvant permettre une cryptanalyse physique des circuits sécurisés d'une part et la conception de contre-mesures matérielles efficaces pour empêcher ou détecter ces injections d'autre part, est un vaste sujet de recherche.

L'objectif de cette thèse est d'étudier dans un premier temps les moyens d'injection de fautes pouvant conduire à des violations de contraintes temporelles et permettant une cryptanalyse physique des circuits sécurisés. Dans un second temps, d'étudier l'efficacité d'un détecteur conçu pour protéger les circuits contre les injections de fautes par violation des contraintes temporelles et également d'étudier la possibilité d'apparition de nouveaux chemins d'attaque liés à ce détecteur.

Le premier chapitre de ce manuscrit présente succinctement le contexte général de cette thèse. Ensuite, l'algorithme de chiffrement AES qui sert de cible est détaillé. Les techniques de cryptanalyse physique par perturbation et par observation les plus classiques sont présentées. Enfin le fonctionnement des circuits synchrones et les contraintes temporelles qui leurs sont associées sont rappelés.

Dans le second chapitre, nous nous concentrons sur l'étude de l'injection de fautes par violation de ces contraintes temporelles. D'une part, la cible matérielle sur laquelle est implémenté l'algorithme de chiffrement AES est présentée. D'autre part, les bancs d'injection de fautes utilisés sont décrits. Enfin, les implémentations matérielles d'un voltmètre intégré et d'un détecteur visant à prévenir les violations de contraintes temporelles sont présentées.

Dans le chapitre 3, les techniques d'injection de fautes par violation de contraintes temporelles sont effectivement mise en œuvre. Dans un premier temps des attaques statiques (permanentes) sont menées : par augmentation de la fréquence ou de la température et par diminution de la tension d'alimentation. Les résultats ainsi obtenus sont comparés pour confirmer qu'il s'agit bien de violations de contraintes temporelles relatives au temps de setup. Dans un second temps des attaques dynamiques (transitoires) sont menées : par augmentation de la fréquence et variations transitoires de la tension (positives et négatives). Les résultats obtenus sont à nouveau comparés pour confirmer qu'il s'agit encore de violations de contraintes sur le temps de setup. Enfin le voltmètre intégré décrit dans le chapitre 2 est utilisé pour observer les perturbations effectivement injectées, dans le circuit soumis à des variations transitoires (glitchs) de tension.

Le chapitre 4 présente l'étude de l'efficacité du détecteur d'injection de fautes par violation de contraintes temporelles soumis à des attaques électromagnétiques. L'objectif est de démontrer l'existence d'un effet local lié à ces impulsions électromagnétiques. Cet effet local pourrait mettre en défaut le détecteur. Une méthode d'amélioration de la protection consistant à implémenter une matrice de détecteurs est ensuite proposée.

Le dernier chapitre met en évidence l'existence d'un nouveau canal auxiliaire lié au couplage électrique entre blocs logiques indépendants. Il est illustré par l'étude de la fuite d'information entre le détecteur et l'implémentation de l'AES-128.

Chapitre 1

État de l'art

Sommaire

La cryptographie est la science relative à la transmission d'informations de façon secrète. L'objectif est de rendre un message incompréhensible aux personnes ne disposant pas du secret cryptographique, une clé le plus souvent. Lui est opposée la cryptanalyse qui est l'art de retrouver par l'analyse mathématique ou physique toute information inintelligible ou secret cryptographique. Ces deux disciplines forment ensemble la cryptologie ou la science du secret.

1.1 Algorithmes de chiffrement

1.1.1 Introduction historique

Un des premiers exemples de communication se voulant sécurisé est celui des Scytales. Cette technique de chiffrement par transposition utilise un bâton de diamètre

choisi (le secret cryptographique) appelé scytale. Une bandelette de cuir s'enroule autour de la scytale et un message y est inscrit. Le principe de cette technique est donc de changer la position des lettres dans le message. Une fois la bandelette déroulée, il faut posséder une scytale de diamètre identique et y enrouler la bandelette pour pouvoir déchiffrer le message. Une autre technique très connue est le chiffre de César [Kahn 2008]. Il s'agit de la méthode cryptographique par substitution mono-alphabétique la plus ancienne (I^{er} siècle av. J.C.). Le principe est de remplacer chaque lettre d'un message clair par la n^{eme} lettre suivante de l'alphabet. D'autres algorithmes de chiffrement sont apparus plus tard. Cependant, la première guerre mondiale marque l'avènement de l'usage de la cryptographie. En effet une bonne maîtrise de la cryptanalyse permettait d'avoir un avantage considérable sur l'ennemi. La seconde guerre mondiale, à travers l'utilisation de machines mécaniques à chiffrer (la machine Enigma) marque les débuts de l'automatisation de la cryptographie. La machine Enigma est une machine à chiffrer électromécanique inventée en 1919 par un ingénieur allemand. Aujourd'hui les algorithmes de chiffrement sont utilisés dans de nombreuses applications souvent critiques et notamment dans de nombreux systèmes électroniques embarqués. Ces algorithmes permettent l'échange d'informations chiffrées via un canal non sécurisé et répondent à des propriétés de confidentialité, intégrité, authentification et non répudiation.

1.1.2 Algorithmes de chiffrement symétriques

Pour les algorithmes symétriques, la clé publique et la clé privée sont identiques (voir la figure 1.1). Cette clé unique et secrète est connue seulement du destinataire et de l'expéditeur des données confidentielles. Elle doit évidemment rester inconnue pour les tiers.

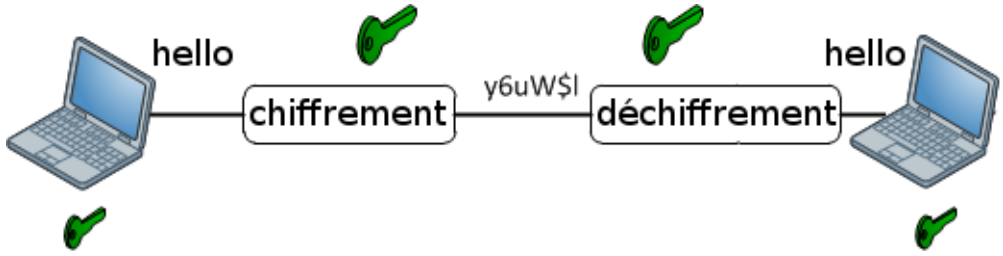

FIGURE 1.1 – Principe d'un protocole de chiffrement symétrique

Ces algorithmes se déclinent en deux catégories principales : les chiffrements

par flot et les chiffrements par blocs. Les chiffrements par flot tel que A5/1, E0 ou encore RC4 arrivent à traiter les données de longueur quelconque. Dans le cadre d'un chiffrement par blocs les données sont d'abord séparées en blocs de n bits (le plus souvent 32, 64, 128 ou 512 bits). Ensuite chacun de ces blocs est chiffré séparément par l'algorithme puis sont rechaînés pour former le message chiffré. Certains de ces algorithmes largement répandus tels que DES [NIST 1999] ou AES [NIST 2001] sont des standards de chiffrement définis par le NIST (National Institute of Standards and Technology).

1.1.3 Algorithmes de chiffrement asymétriques

Le principe de chiffrement/déchiffrement asymétrique, illustré par la figure 1.2, a été introduit en 1976 par Diffie et Hellman [Diffie 1976]. La cryptographie asymétrique repose sur l'existence de fonctions mathématiques à sens unique qui une fois appliquées à un message le rendent extrêmement difficile à déchiffrer. Les algorithmes asymétriques reposent sur l'utilisation de deux clés distinctes liées mathématiquement entre-elles : une pour le chiffrement et une pour le déchiffrement. La clé de chiffrement est appelée clé publique et n'importe qui peut l'utiliser pour générer un message chiffré. La clé de déchiffrement dite privée, quant à elle, n'est connu que du destinataire. Seul ce dernier peut déchiffrer le message.

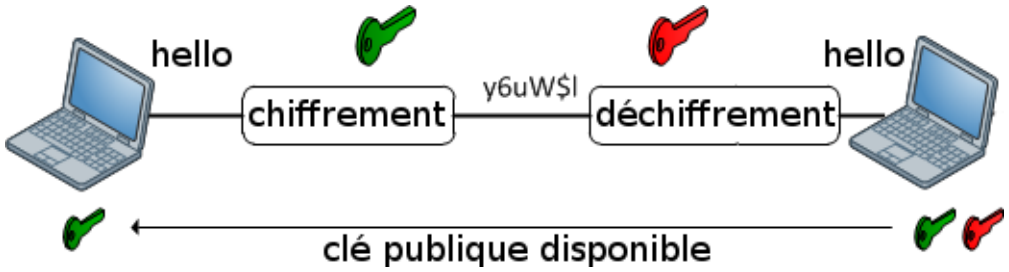

FIGURE 1.2 – Principe d'un protocole de chiffrement asymétrique

Les algorithmes asymétriques sont le plus souvent utilisés pour échanger des clés à usage unique qui seront ensuite utilisées pour des chiffrements symétriques. Aujourd'hui le RSA [Rivest 1978] est l'algorithme asymétrique le plus connu.

1.1.4 Description du standard de chiffrement symétrique AES

AES est le standard de chiffrement symétrique actuel recommandé par le NIST. Dans ce travail, une implémentation matérielle de cet algorithme servira de cible

pour comparer différents moyens d'injection de fautes. Il s'agit d'un réseau d'opérations de substitutions et de permutations [Stinson 2005] basé sur 4 opérations de base appelées : SUBBYTE, SHIFTROWS, MIXCOLUMN, ADDROUNDKEY. Ces 4 étapes sont utilisées de façon itérative sous forme de "rondes" comme illustré dans la figure 1.3. Le nombre de rondes de cet algorithme (10, 12 ou 14 après une ronde initiale) dépend de la taille de clé (respectivement, 128, 192 ou 256 bits). Dans le cadre de cette thèse seules des clés de 128 bits ont été utilisées. Les données manipulées sont le plus souvent représentées par une matrice de 4×4 octets appelée matrice d'état.

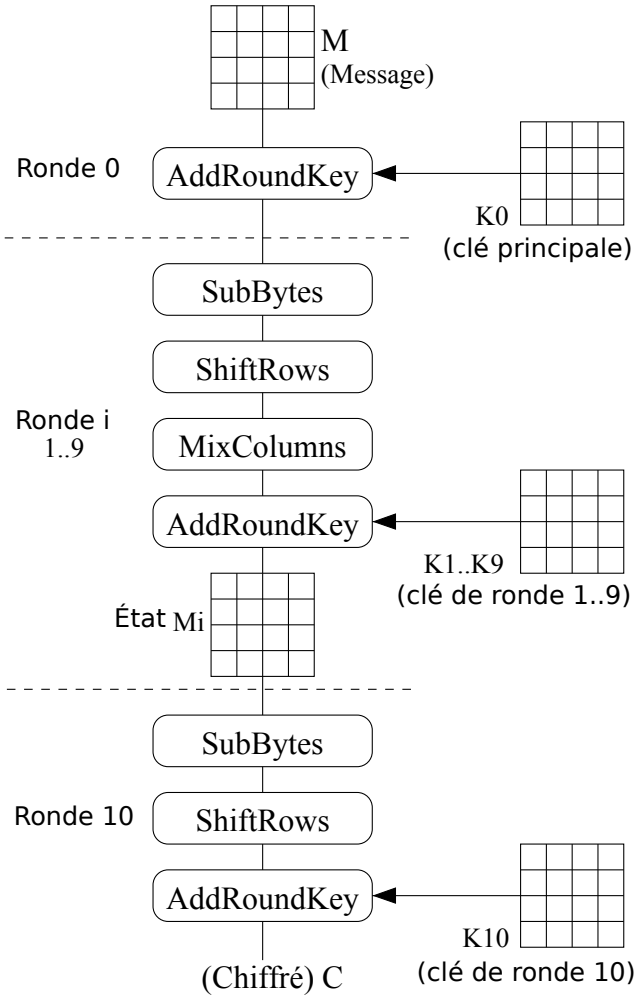

FIGURE 1.3 – Algorithme de chiffrement AES-128

Mi est la matrice d'état à la fin de la i^{eme} ronde de l'AES.
M est le texte clair en début de chiffrement

C est le résultat du chiffrement.
K est la clé de chiffrement.
Ki est la sous-clé utilisée à la i^{eme} ronde de l'AES dérivée de la clé K. (voir la figure 1.3).

- L'opération SUBBYTE est une transformation non linéaire appliquée à chacun des octets de la matrice d'état. Cette opération de substitution utilise une table de substitution appelée S-Box transformant un octet en un autre.
- L'opération SHIFTROWS est un décalage à gauche des octets de chaque ligne d'une matrice d'état. La première ligne n'est pas affectée, la deuxième ligne est décalée d'un rang vers la gauche, la troisième ligne de deux rangs et enfin la dernière ligne de la matrice d'état est décalée de trois rangs.
- La transformation MIXCOLUMN réalise une multiplication matricielle dans le corps de Galois $GF(2^8)$ de la matrice d'état. Chaque colonne de la matrice d'état est multipliée par la matrice donnée figure 1.4 :

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix}$$

FIGURE 1.4 – Matrice de multiplication du MixColumn

- L'opération ADDROUNDKEY est un "ou exclusif" effectué octet par octet entre la matrice d'état et la clé de ronde.

Les sous-clés de ronde sont calculées à partir de la clé principale par un module appelé KEY EXPANDER de la façon suivante :

$K0$, la sous-clé de la ronde 0 représente la clé principale. Pour les autres sous-clés, Ki avec $i \in [1..10]$, plusieurs opérations sont utilisées. La première est une rotation de bas en haut de la dernière colonne des sous-clés. L'opération SUBBYTE précédemment introduite est réutilisée. Si est la valeur en sortie de la table de substitution (S-Box) pour une entrée Ki. Et enfin, un "ou exclusif" effectué octet par octet est aussi utilisé avec la matrice RCON (constante de ronde définie par le standard) présentée dans la figure 1.5.

$$\begin{pmatrix} 01 & 02 & 04 & 08 & 10 & 20 & 40 & 80 & 1B & 36 \end{pmatrix}$$

FIGURE 1.5 – Matrice RCON pour un AES-128

Le 4 premiers octets de la i^{eme} sous-clé ($i \in [1..10]$) sont calculés comme suit :
$Ki_1 = K(i-1)_1 \oplus S(i-1)_{14} \oplus RCON(i)$
$Ki_2 = K(i-1)_2 \oplus S(i-1)_{15}$
$Ki_3 = K(i-1)_3 \oplus S(i-1)_{16}$
$Ki_4 = K(i-1)_4 \oplus S(i-1)_{13}$
Et ensuite les 12 autres octets de la i^{eme} sous-clé (notés Ki_j avec $j \in [5..16]$) sont calculés comme suit :
$Ki_j = K(i-1)_j \oplus Ki_{j-4}$

1.2 Introduction aux techniques de cryptanalyse physique à l'aide d'attaques non-invasives

Les algorithmes cryptographiques utilisés sont supposés sûrs mathématiquement. Aussi, une cryptanalyse mathématique de ces algorithmes est a priori impossible avec les puissances de calculs actuelles. Cependant leurs implémentations physiques, notamment en technologie CMOS, génèrent des grandeurs observables et peuvent être perturbées. Celles-ci permettent de réaliser des attaques avec pour objectif de retrouver l'intégralité ou une partie des informations secrètes : elles sont appelées attaques par canaux auxiliaires (Side Channels Analysis, SCA). Ces attaques physiques sont généralement conduites en 2 étapes :

- Une étape d'extraction d'informations. Cette étape peut être menée à l'aide de perturbations (attaques actives) ou à l'aide d'observations (attaques passives) du circuit.
- Une étape de traitement des données pour en tirer des renseignements sur le secret cryptographique, il s'agit alors de cryptanalyse physique.

Ces attaques sont classifiées en deux catégories :

- Non-invasives : ce sont des attaques ne nécessitant aucune préparation du circuit. Il peut s'agir d'une simple observation de la consommation d'énergie [Kocher 1999], de l'émanation électromagnétique [Gandolfi 2001] ou du temps d'exécution [Koeune 2005] par exemple. On parlera dans ce cas d'at-

taques passives. Il peut aussi s'agir d'attaques dites actives ou attaques en fautes (fault attacks, FA) qui perturbent le fonctionnement normal du système en utilisant, par exemple, des variations statiques ou transitoires (glitchs) de la fréquence d'horloge [Agoyan 2010b], de la tension [Barenghi 2010] [Selmane 2008] , de la température ou de l'environnement électromagnétique [Dehbaoui 2012a]. Les deux types d'attaques, actives et passives, peuvent être combinées [Roche 2011].

– Invasives ou semi-invasives : elles nécessitent des modifications plus ou moins importantes du circuit. Ces modifications peuvent être partielles et non destructives permettant ensuite l'utilisation d'équipements de mesure (ou de "probing") [Handschuh 1999] [Gammel 2010] donnant accès à des informations critiques (des valeurs intermédiaires de calcul par exemple). Elles peuvent être superficielles ne nécessitant qu'une altération du boîtier protégeant le circuit sans modifier le circuit lui-même. L'ouverture du boîtier permet par exemple l'injection de fautes par laser [Skorobogatov 2003] [Agoyan 2010a], l'observation de points chauds ou d'émission de photons [Schlosser 2012] ou d'améliorer l'efficacité des injections de impulsions électromagnétiques ainsi que les observations d'émissions électromagnétiques. Enfin, les modifications peuvent être destructives avec pour objectif de caractériser l'implémentation matérielle couche par couche (rétro-conception) [Kömmerling 1999].

Certaines attaques ne visent pas directement les algorithmes cryptographiques. Une attaque peut avoir pour objectif de perturber le fonctionnement du système plus globalement, en évitant une vérification de code Pin, en empêchant l'exécution de certaines opérations [Moro 2013] ou encore en modifiant directement des informations en mémoire. Dans ce cas, il ne s'agit plus de cryptanalyse. Le "Reset Glitch Hack" est un exemple récent d'attaque concrète ne visant pas un algorithme de chiffrement mais la vérification de signature sur XBox permettant ainsi l'exécution de jeux copiés [DeBusschere 2012].

Toutes ces attaques n'induisent pas les mêmes effets sur le circuit cible. Dans le cadre de cette thèse nous avons étudié les attaques pouvant entraîner des violations de contraintes temporelles. Ces contraintes seront présentées section 1.3.

1.2.1 Cryptanalyse passive

Pour retrouver la clé suite à une observation passive, l'attaquant s'appuie sur les fuites du système (radiations électromagnétiques, consommation de courant, chaleur,

etc.) qui dépendent du secret cryptographique manipulé pendant le chiffrement. En effet, les portes logiques changeant d'état consomment plus que celles dont l'état reste stable. Il existe donc une corrélation entre les données manipulées et la consommation électrique. Dès lors, si un algorithme cryptographique est implémenté en technologie CMOS, l'étude de la consommation de courant et/ou d'autres fuites physiques peut permettre de retrouver le secret cryptographique ou une partie de celui-ci.

La première attaque basée sur l'analyse de la consommation de courant a été introduite par Kocher (Differential Power Analysis, DPA, [Kocher 1999]). L'étude des radiations électromagnétiques peut être une alternative à l'observation de la consommation de courant [Gandolfi 2001]. Plus tard, l'analyse d'une corrélation entre la consommation réelle (mesures pratiques) et un modèle de fuite (hypothèses théoriques) sera introduit (Correlation Power Analysis, CPA, [Brier 2004]). En 2008, l'analyse de l'information mutuelle (Mutal Information Analysis, MIA, [Gierlichs 2008]) a été présentée. Elle a l'avantage de mettre en évidence n'importe quelle relation entre deux variables (les mesures et les hypothèses utilisées dans une CPA par exemple) et donc ne nécessite pas une caractérisation préalable de la fuite. Un autre type d'attaque (Template Attack, TA, [Chari 2003]) permet, sous certaines conditions, de retrouver la clé avec un nombre limité de traces. Cependant pour utiliser cette méthode l'attaquant doit posséder un système identique à celui qu'il analyse sur lequel il a tous les accès afin de pouvoir caractériser le bruit et construire une bibliothèque de traces qu'il comparera ensuite au système qu'il veut attaquer.

Des contre-mesures ont donc été conçues pour prévenir des attaques passives telle que le dual-rail avec précharge de la logique (DPL). Le DPL a pour objectif de rendre la consommation aussi uniforme que possible. D'autres contre-mesures telles que le masquage qui a pour objectif de décorréler les données sensibles et les grandeurs observables (consommation électrique, radiations électromagnétiques, etc.) sont traitées plus en détail dans [Mangard 2010].

La DPA et la CPA étant évoquées dans la suite de cette thèse (voir le chapitre 5), ces attaques sont décrites plus en détail dans les sous-sections suivantes.

1.2.1.1 Analyse différentielle de la consommation (Differential Power Analysis, DPA)

La DPA [Kocher 1999] est une attaque par observation (passive). L'attaquant doit observer le comportement du circuit pour en déduire des informations sur le secret cryptographique. Dans le cadre d'une DPA, l'attaquant récupère dans un premier temps des courbes de consommation de courant (appelées traces). Dans un second temps l'attaquant fait l'hypothèse que la consommation de courant dépend, dans une certaine mesure, des données manipulées par l'algorithme cryptographique. Alors en triant les courbes selon une hypothèse faite sur le secret cryptographique et en comparant les groupes obtenus après le tri, il est possible de vérifier si cette hypothèse est correcte ou non. Ces dépendances ont tendance à être petites et souvent indiscernables a priori parmi le bruit et les erreurs de mesures. Cependant si dépendance il a, il est alors possible de retrouver des informations en considérant des données statistiques obtenues avec un grand nombre d'observations.

Pour mener à bien une attaque par observation, il faut pouvoir récupérer une observable (ici, la consommation de courant) pour des entrées différentes et un secret cryptographique identique. Dans le cas de l'AES il faut pouvoir mesurer la consommation de courant pour une même clé secrète K (inconnue de l'attaquant) et plusieurs textes clairs (ou messages) M_n avec $n \in [1..n_{max}]$ (connus de l'attaquant). Il obtient donc n_{max} traces, $T_{1..n_{max}}[1..t_{max}]$ avec t_{max} le nombre d'échantillons de mesure de la trace.

L'attaquant fait ensuite l'hypothèse que la consommation du circuit dépend à un instant t donné de la valeur d'un bit particulier, $V[b]$. Dans l'équation 1.1, α est la consommation de courant liée au bit considéré et β est la consommation liée à tout le reste du circuit.

$$T[t] = \alpha \times V[b] + \beta \tag{1.1}$$

L'attaquant doit alors simuler l'algorithme en utilisant les mêmes messages M_n et en faisant des hypothèses, K_h, sur la clé secrète. Il obtient ainsi des valeurs intermédiaires de calcul dépendant de l'hypothèse faite sur la clé secrète. L'attaquant peut alors utiliser la valeur hypothétique de $V[b]$ et la comparer à la valeur observée $T[t]$ pour confirmer ou infirmer la validité de ses hypothèses. Cette valeur hypothétique est appelée la *fonction de sélection*.

Cette fonction de sélection, comme il s'agit de la valeur d'un bit dans le cas de la DPA, peut valoir '0' ou '1' en fonction du message, M_n, et de l'hypothèse de clé,

K_h. $V[b](M_n, K_h) = 0$ ou 1. Pour chaque hypothèse de clé, K_h, il est donc possible de diviser les traces observées en deux "paquets" selon la valeur de la fonction de sélection.

- Si $V[b](M_n, K_h) = 0$, alors T_n est dans le paquet "0"
- Si $V[b](M_m, K_h) = 1$, alors T_m est dans le paquet "1".

Enfin, pour chaque hypothèse de clé, K_h, la moyenne des traces de chaque paquet est calculée, $T0_{moy}[t]_{K_h}$ et $T1_{moy}[t]_{K_h}$ et la différence de ces moyennes est calculée : $\Delta_{moy}[t]_{K_h}$.

$$T0_{moy}[t]_{K_h} = \frac{\sum_{n=1}^{n_{max}}(1 - V[b](M_n, K_h)) \times T_n[t]}{\sum_{n=1}^{n_{max}}(1 - V[b](M_n, K_h))} \tag{1.2}$$

$$T1_{moy}[t]_{K_h} = \frac{\sum_{n=1}^{n_{max}}(V[b](M_n, K_h)) \times T_n[t]}{\sum_{n=1}^{n_{max}}(V[b](M_n, K_h))} \tag{1.3}$$

$$\Delta_{moy}[t]_{K_h} = T1_{moy}[t]_{K_h} - T0_{moy}[t]_{K_h} \tag{1.4}$$

Si l'hypothèse de clé est fausse, la valeur du bit calculé théoriquement ne correspond pas à la valeur effective du bit. Dans ce cas la différence des moyennes tends vers zéro (voir l'équation 1.5) :

$$\forall t \in [1..t_{max}], \lim_{n_{max} \to \infty} \Delta_{moy}[t]_{K_{incorrect}} \approx 0 \tag{1.5}$$

Par contre si l'hypothèse de clé est correcte le différences des moyennes sera différente de zéro (voir l'équation 1.6) :

$$\exists t \in [1..t_{max}], \lim_{n_{max} \to \infty} \Delta_{moy}[t]_{K_{correct}} \neq 0 \tag{1.6}$$

La consommation de courant n'est pas dépendante de la valeur d'un bit en particulier (b) tout au long du chiffrement. Aussi $\Delta_{moy}[t]_{K_{correct}}$ ne sera pas très différent de zéro sur toute la durée des traces acquises. Le plus souvent, des "pics" sont observées aux instants où les calculs internes de l'algorithme sont dépendants du bit considéré dans la fonction de sélection.

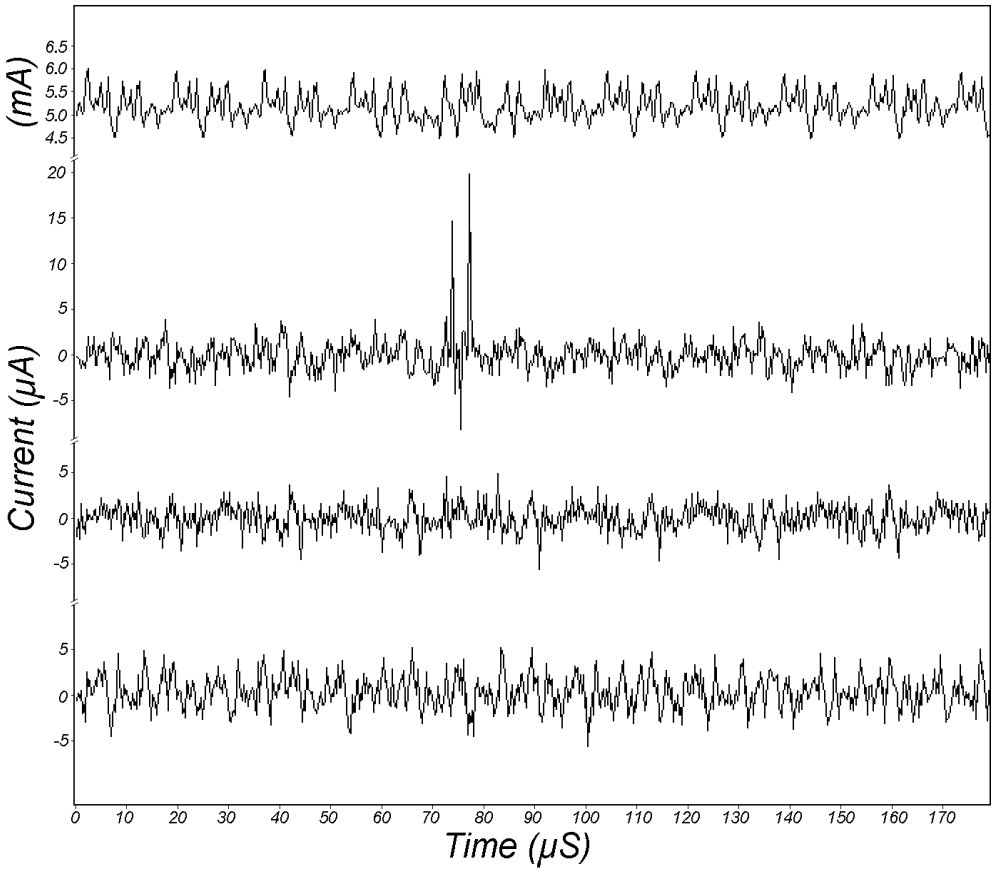

FIGURE 1.6 – Traces de DPA, 1 correcte et 2 incorrectes, avec la référence de consommation, image tirée de [Kocher 1999]

À titre d'exemple, la figure 1.6 tirée de [Kocher 1999] présente la moyenne de la consommation pour l'ensemble des chiffrements effectués (trace du haut). En dessous la figure présente 3 différences des moyennes obtenues avec des messages connus chiffrés par une implémentation de DES. Dans le premier cas, l'hypothèse est correcte et dans les deux cas suivants les hypothèses sont incorrectes. Dans le cas où l'hypothèse est correcte, des pics sont observés.

1.2.1.2 Analyse par corrélation de la consommation (Correlation Power Analysis, CPA)

La *corrélation de Pearson* peut être utilisée comme une alternative à la DPA. La corrélation de Pearson renvoie le taux de corrélation linéaire entre deux variables. Dans notre cas, nous nous intéressons au taux de corrélation linéaire entre les

mesures (traces T_n) et la fonction de sélection qui dépend des hypothèses de clé K_h. Cette attaque est très similaire à la DPA mais cette fois des fonctions de sélection plus complexes peuvent être utilisées. Dans la CPA classique, c'est le poids de Hamming (Hamming weight, HW) d'un octet, $HW[o]$, qui est utilisé comme fonction de sélection à la place de la valeur d'un seul bit. Le poids de Hamming représente le nombre de bits à l'état haut d'un mot binaire considéré. Aussi l'équation 1.1 devient l'équation 1.7 :

$$T[t] = \alpha \times HW[o] + \beta \tag{1.7}$$

La CPA consiste à calculer pour chaque hypothèse de clé secrète K_h le coefficient de la corrélation de Pearson entre la mesure à un instant t, et le poids de Hamming de l'octet considéré, $HW[o]$, prédit en fonction de l'hypothèse K_h : $\rho(T_n[t], HW[o](M_n, K_h))$ (voir l'équation 1.8). Par soucis de simplification $HW[o](M_n, K_h)$ sera noté simplement H_{M_n,K_h}.

$$\rho_{K_h} = \frac{n_{max}(\sum T_n[t] \times H_{M_n,K_h}) - (\sum T_n[t])(\sum H_{M_n,K_h})}{\sqrt{(n_{max}\sum T_n[t]^2 - (\sum T_n[t])^2)(n_{max}\sum H_{M_n,K_h}^2 - (\sum H_{M_n,K_h})^2)}} \tag{1.8}$$

Dans l'équation précédente 1.8 $\sum$ est définit comme étant égal à $\sum_{n=1}^{n_{max}}$.

L'équation 1.8 représente ρ_{K_h}, le coefficient de corrélation entre les mesures et les prédictions faites pour une hypothèse de clé, K_h. n_{max} est le nombre de messages qui ont été chiffrés pour observer les traces T_n avec $n \in [1..n_{max}]$. Si la fonction de sélection choisie est la valeur d'un seul bit, alors les résultats obtenus sont très similaires à ceux obtenus avec une DPA.

1.2.2 Cryptanalyse active

Pour retrouver la clé secrète suite à une attaque active, l'attaquant exploite les différences induites par l'attaque entre une sortie correcte et une sortie incorrecte du système (Differential Fault Analysis, DFA, [Boneh 1997] [Biham 1997]). Des contre-mesures telles que la redondance temporelle [Maistri 2007] ou spatiale [Joye 2007], [Di Natale 2007] de l'algorithme ont été introduites pour prévenir des attaques de type DFA. [Maistri 2011] récapitule la plupart de ces contre-mesures. Par la suite de nouveaux chemins d'attaques sont apparus. Les attaques dites en Safe Error [Yen 2000] [Blömer 2003] s'appuient sur le fait que la valeur d'un bit puisse être forcée avec certitude sans nécessairement conduire à un résultat de calcul fauté. L'attaquant peut alors déduire de l'absence de faute que la valeur forcée

était la valeur de ce bit avant l'attaque. Les informations secrètes peuvent aussi être extraites de la différence de comportement du circuit soumis à la perturbation (Differential Behavior Analysis, DBA, [Robisson 2007]). Une application du principe de la DBA, publiée récemment, repose sur l'analyse de la sensibilité du circuit soumis à une attaque physique (Fault Sensitivity Analysis, FSA, [Li 2012]). Pour se protéger de ce dernier type d'analyse, des contre-mesures ont été conçues de façon à être logiquement indépendantes de l'algorithme qu'elles protègent [Selmane 2011], [Endo 2012]. En d'autres termes, les données traitées par l'algorithme n'influent théoriquement pas sur la détection ou non d'une attaque.

La DFA et la FSA étant évoquées dans la suite de cette thèse (voir respectivement les chapitres 3 et 5), ces attaques sont décrites plus en détail dans les sous-sections suivantes.

1.2.2.1 Analyse différentielle de fautes (Differential Fault Analysis, DFA)

Depuis la publication de la DFA en 1997 [Boneh 1997] [Biham 1997] de nombreuses attaques utilisent cette méthode d'analyse. Le principe de la DFA est de comparer un chiffré fauté, D, et un chiffré non fauté, C, afin d'extraire des informations sur le secret cryptographique. Pour réaliser ce type d'attaque, une très bonne maîtrise de la perturbation injectée est nécessaire. Si la faute réellement injectée correspond bien au modèle considéré, seul un certain nombre de clés sont cohérentes avec l'erreur, $e = D \oplus C$, obtenue en sortie. En reproduisant l'attaque plusieurs fois le nombre de clés cohérentes avec les erreurs obtenues diminue. Parfois les opérations de ronde telles que le SUBBYTES [Giraud 2005] [Lashermes 2012] ou le MIXCOLUMNS [Piret 2003] [Moradi 2006] sont ciblées, mais aussi, dans d'autres cas, le calcul des sous-clés [Roche 2011] [Kim 2012] ou encore la réduction du nombre de rondes [Tunstall 2005] [Dutertre 2012]. Les exemples suivants sont des applications théoriques de la DFA. Ils sont présentés pour illustrer l'importance qu'une bonne maîtrise de l'instant d'injection ainsi que la maîtrise de l'étendue des fautes (1 octet, 1bit) sont primordiales.

Dans les équations et les figures suivantes, les opérations de SUBBYTES seront notées SB, les opérations de SHIFTROW seront notées SR et les opérations de ADDROUNDKEY seront notées ARK.

Attaque de Giraud

L'attaque décrite dans [Giraud 2005] utilise des fautes mono-bit devant être injectées avant la transformation SUBBYTES de la dernière ronde. Le modèle de fautes de cette attaque est donc d'obtenir des fautes mono-bit sur un ou plusieurs octets de la matrice d'état en début de dernière ronde. La figure 1.7 présente un schéma de la dernière ronde de l'algorithme AES lorsqu'une faute, octet marqué en rouge, est injectée sur un octet de la matrice d'état avant l'opération SUBBYTES. On remarque que la faute reste confinée à un seul octet en raison de l'absence de la transformation MIXCOLUMNS durant la dernière ronde. Pour retrouver la valeur de l'octet de $K10$ correspondant à la localisation de la faute injectée, on compare alors le chiffré correct avec le chiffré fauté.

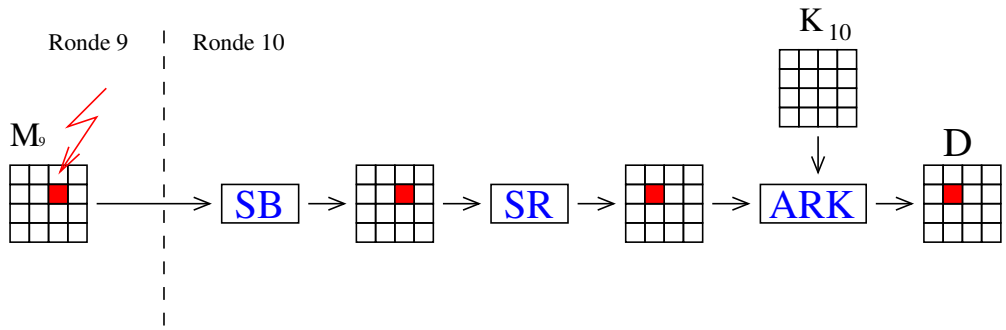

FIGURE 1.7 – Illustration de l'attaque Giraud mono-bit

L'équation 1.9 représente les différentes opérations de la dernière ronde de l'AES non fauté :

$$C = SR \circ SB(M9) \oplus K10 \tag{1.9}$$

Les opérations de la dernière ronde étant effectuées octet par octet, pour plus de simplicité, la transformation SHIFTROWS peut être omise sans altérer la portée de l'analyse. On obtient une équation simplifiée pour C (équation 1.10) :

$$C = SB(M9) \oplus K10 \tag{1.10}$$

Pour le chiffré fauté, D, on obtient (équation 1.11) :

$$D = SB(M9 \oplus E9) \oplus K10 \tag{1.11}$$

En appliquant un OU-EXCLUSIF entre les équations 1.10 et 1.11, on obtient suc-

cessivement les équations 1.12 et 1.13 :

$$\Delta = C \oplus D \tag{1.12}$$

$$\Delta = SB(M9) \oplus SB(M9 \oplus E9) \tag{1.13}$$

L'attaque repose sur le fait que les fautes injectées sont mono-bit, dès lors il existe 2040 (255×8) couples $(M9, E9)$ possibles. Pour chaque couple possible, Δ est calculé puis comparé avec la valeur de Δ trouvée expérimentalement pour le couple de chiffrés correct/fauté. On obtient alors un ensemble de valeurs possibles pour $(M9, E9)$ contenant la valeur correcte pour $M9$. Si cet ensemble contient plusieurs valeurs, on les teste alors de la même manière avec un second couple de chiffrés correct/fauté. L'opération est recommencée avec d'autres couples de textes chiffrés correct/fauté jusqu'à ce que l'ensemble des valeurs possibles pour $(M9, E9)$ ne contienne plus qu'une seule possibilité. Ayant trouvé la valeur de $M9$, l'équation 1.14 permet de calculer la valeur de $K10$.

$$K10 = SB(M9) \oplus C \tag{1.14}$$

Une fois que tous les octets de K10 sont connus, pour retrouver la clé K, il suffit alors de faire l'opération de calcul inverse des clés de ronde.

Selon [Giraud 2005], cette attaque est efficace à 97% avec trois couples de chiffrés correct-fauté (C, D) pour retrouver un octet de la sous-clé $K10$.

Attaque de Roche et al.

Cette attaque vise le module de calcul des clés de ronde de l'algorithme. Une faute doit être injectée sur la clé de l'avant-dernière ronde (ronde 9) de sorte que $K9$ et $K10$ soient affectées par cette faute, on a alors :

$$\tilde{K9} = K9 \oplus E9 \tag{1.15}$$

$$\tilde{K10} = K10 \oplus E10 \tag{1.16}$$

où $\tilde{K9}$ et $\tilde{K10}$ sont les deux dernières clés de ronde fautées. $E9$ et $E10$ sont les valeurs des fautes des deux dernières clés de ronde. L'attaque présentée en 2011 [Roche 2011] se déroule alors en plusieurs étapes. On commence par chiffrer N messages différents sans injection de fautes. On recommence ensuite avec les mêmes messages mais cette fois-ci une faute est injectée lors du calcul de la sous-clé de la ronde 9. On obtient N couples de chiffrés correct/fauté (C,D). L'analyse de ces

couples chiffrés correct/fauté étant réalisée octet par octet, l'opération SHIFTROWS peut être omise. Les équations 1.17 et 1.18 représentent les calculs de la dernière ronde pour C et D.

$$C = SB(M9) \oplus K10 \tag{1.17}$$

$$D = SB(M9 \oplus E9) \oplus K10 \oplus E10 \tag{1.18}$$

À partir des équations 1.17 et 1.18 on peut exprimer D avec l'équation 1.19 :

$$D = SB(SB^{-1}(C \oplus K10) \oplus E9) \oplus K10 \oplus E10 \tag{1.19}$$

Pour chaque hypothèse possible du triplet ($E9$, $E10$, $K10$) notée ($e9$, $e10$, $k10$), un compteur T est associé à l'octet correspondant. Pour chaque couple de chiffrés correct/fauté, on incrémente la valeur du compteur T associé à chaque triplet si l'équation 1.20 est vérifiée.

$$SB(SB^{-1}(C \oplus k) \oplus E9) \oplus k \oplus e10 = D \tag{1.20}$$

Si les fautes injectées sur la sous-clé de la ronde 9 sont reproductibles, les valeurs de $E9$ et $E10$ sont constantes pour tous les couples de chiffrés correct/fauté. Il n'y a alors qu'un seul triplet ($e9$, $e10$, $k10$) dont la valeur du compteur est N. Il correspond aux valeurs correctes de ($E9$, $E10$, $K10$). Dans le cas où les fautes injectées sont constantes, il faut alors trois couples de chiffrés correct/fauté pour obtenir un taux de réussite de 90%. En revanche lorsque les fautes injectées ne sont pas constantes, le nombre de couples chiffrés correct/fauté pour réaliser l'attaque va alors augmenter en fonction de la variabilité des fautes injectées. Comme le montre la figure 1.8 extraite de [Roche 2011], lorsque le taux de reproductibilité de la faute injectée décroit, le nombre de couples de chiffrés correct/fauté nécessaires pour réussir l'attaque va alors augmenter exponentiellement.

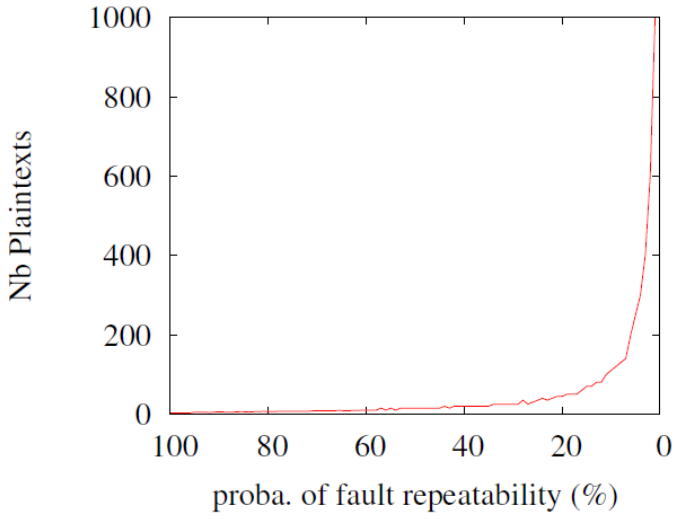

FIGURE 1.8 – Nombre de couples chiffrés correct/fauté nécessaires pour un taux de réussite de 90% en fonction du taux de répétabilité des fautes injectées [Roche 2011].

Pour cette attaque, le modèle de fautes utilisé est différent de l'attaque de Giraud présentée précédemment. Le modèle de fautes est donc d'obtenir une faute sur un ou plusieurs octets lors du calcul de la sous-clé de la ronde 9. De plus, les fautes injectées lors de plusieurs calculs de chiffrement doivent être d'une valeur la plus constante possible, l'idéal étant d'avoir des valeurs de fautes constantes dans 100% des cas. Cette dernière contrainte sur le modèle de fautes implique d'avoir une certaine maîtrise sur les bits fautés.

Attaque de Piret

Cette attaque présentée en 2003 par Piret et al. [Piret 2003], impose des contraintes moins fortes sur le type de fautes que l'attaque sur le SUBBYTES présentée dans [Giraud 2005]. En effet, la faute doit ici être mono-octet et injectée avant la transformation MIXCOLUMNS de la ronde 9. Lorsqu'une faute est injectée en respectant ces contraintes, on obtient un schéma de propagation de la faute à travers l'algorithme de chiffrement similaire à la figure 1.9. L'octet fauté y est marqué en rouge.

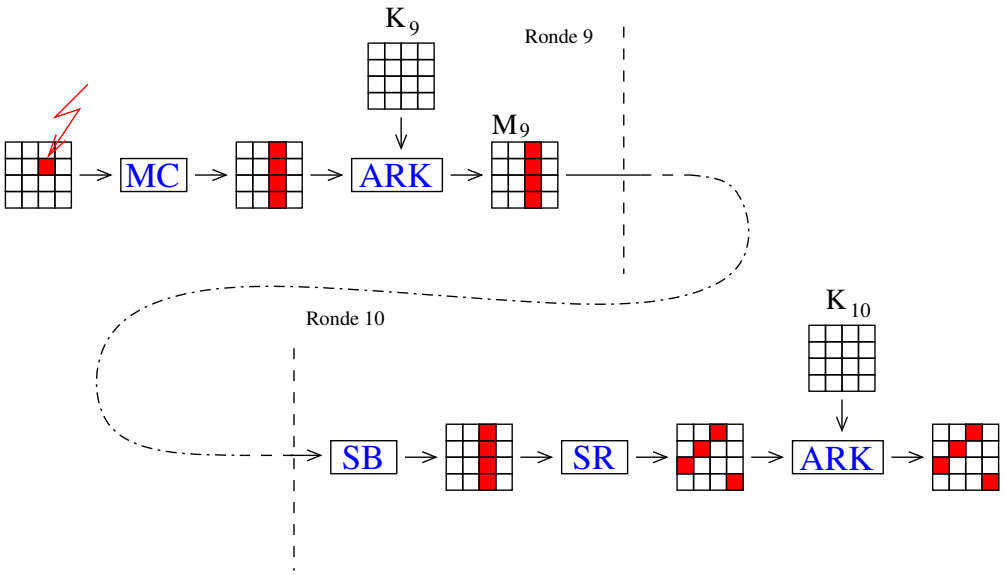

FIGURE 1.9 – Schéma de l'attaque de Piret et al. [Piret 2003].

Cette attaque permet de retrouver quatre octets de la clé de ronde $K10$ avec seulement un octet fauté. Cela est possible grâce à l'opération MIXCOLUMNS qui propage la faute injectée sur tous les octets de la colonne de la matrice d'état, comme on peut l'observer sur la figure 1.9.

La première partie de l'attaque consiste à créer une liste de toutes les valeurs possibles des quatre octets fautés à la fin de le 9^{ieme} ronde. La colonne de la matrice d'état fautée après le MIXCOLUMNS est connue par analyse de la position des quatre octets fautés de D. Néanmoins, la position dans cette colonne de l'octet fauté avant le MIXCOLUMNS est inconnue, on a donc 1020 (4×255) possibilités dans cette liste. On appelle cette liste G. Une seconde liste L contient toutes les valeurs possibles des quatre octets de $K10$ correspondant aux quatre octets fautés. Cependant, pour limiter la complexité des calculs, pour le début de l'analyse, on considère dans la liste L seulement les valeurs possibles de deux octets de $K10$. La liste L contient donc 2^{16} valeurs possibles de deux octets de $K10$ (noté k). Avec C et D, on calcule pour chaque hypothèse de L la valeur E de la faute en début de ronde 10 à partir de l'équation 1.21 :

$$E = SB^{-1}(C \oplus k) \oplus SB^{-1}(D \oplus k) \tag{1.21}$$

Si la valeur de E n'est pas contenue dans G, on retire la valeur k de L et on passe à la valeur suivante. Une fois que toutes les valeurs de L ont été testées, la taille de

l'ensemble L est significativement réduite. On étend ensuite la liste avec le troisième octet de $K10$ et on recommence. On fait ensuite de même avec le quatrième octet. On obtient à la fin les valeurs correctes pour les quatre octets de $K10$ correspondant aux quatre octets fautés.

Pour avoir une efficacité de 99%, deux couples de chiffrés correct/fauté sont nécessaires. Avec cette attaque, on doit donc injecter une faute sur un octet de chaque colonne de la matrice d'état pour pouvoir retrouver les 16 octets de la clé secrète.

Dans ce même article, les auteurs proposent d'injecter une faute, une ronde plus tôt dans le déroulement de l'algorithme. En effet, si une faute est injectée avant la transformation MIXCOLUMNS de la 8^{ieme} ronde, elle va alors se propager sur toute la colonne de la matrice d'état. On aura une faute sur un octet de chaque colonne de la matrice d'état avant le dernier MIXCOLUMNS. A la fin du chiffrement, les 16 octets seront fautés. En appliquant l'analyse présentée précédemment sur chaque colonne, on retrouve alors les 16 octets de la clé de chiffrement.

En injectant une faute avant le MIXCOLUMNS de la 8^{ieme} ronde, 2 couples de chiffrés correct/fauté sont nécessaires pour atteindre un taux de réussite de 77%.

1.2.2.2 Analyse de la sensibilité à l'injection de fautes (Fault Sensitivity Analysis, FSA)

La FSA est une attaque basée sur l'injection de fautes comme la DFA. Cependant le principe de l'attaque est plutôt similaire à celui d'une CPA. Contrairement à une DFA cette attaque ne repose pas sur l'exploitation de la faute injectée (la valeur du chiffré fauté, D) mais sur la sensibilité du circuit à l'injection de fautes (le stress limite à partir duquel la faute est injectée). Le stress appliqué à un circuit est augmenté progressivement jusqu'à ce qu'une valeur critique soit détectée. Ensuite cette valeur de stress critique est corrélée aux données sensibles manipulées par le circuit selon le même protocole que celui utilisé dans une attaque par observation classique.

La FSA se base sur le fait que les temps de propagation dans un arbre de logique combinatoire dépendent des données manipulées. Les auteurs de [Li 2012] proposent une simulation des temps de propagation dans une S-Box (PPRM1) pour confirmer cette dépendance. La figure 1.10 représente la valeur du temps critique de l'opération de SUBBYTE pour un octet ($Tc(o)$) en fonction du poids de Hamming de cet octet ($HW(o)$). Ces résultats confirment qu'il existe une corrélation entre ces

deux grandeurs.

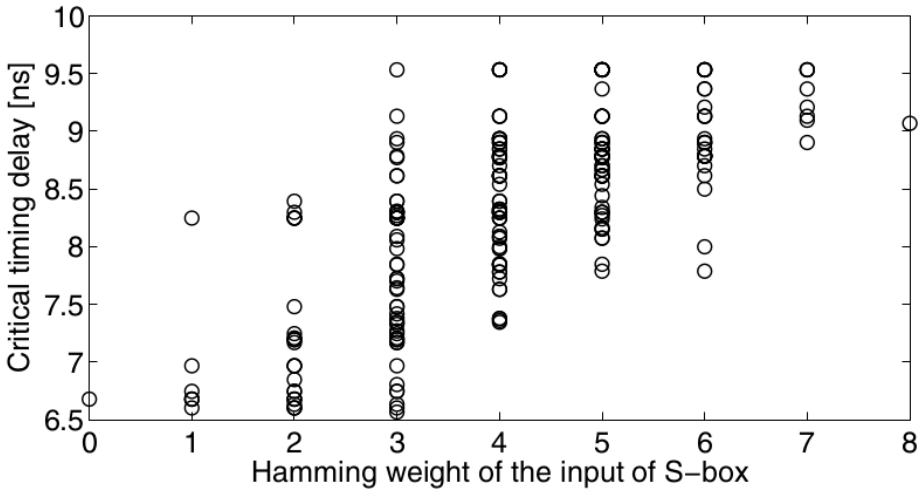

FIGURE 1.10 – Temps critique $Tc(o)$ en fonction de poids de Hamming $HW(o)$ - extrait de [Li 2012]

Le modèle proposé est donc le suivant : pour un message M_n le temps critique dépend linéairement du poids de Hamming de l'octet considéré $HW_n(o)$, comme présenté par l'équation 1.22.

$$Tc_n = \alpha \times HW_n(o) + \beta \tag{1.22}$$

L'attaque se fait en trois étapes similaires à celles présentées pour la DPA : observations, hypothèses, corrélation. D'abord l'attaquant récupère une information observable pour des entrées différentes et un secret cryptographique identique. Dans le cas de la FSA, il ne s'agit pas vraiment d'une observation passive puisque l'attaquant augmente progressivement le stress appliqué au circuit jusqu'à ce qu'une faute soit injectée. Il récupère ainsi la valeur du temps critique pour différentes entrées. L'attaquant obtient donc après cette première étape, des couples {Message (M_n), Temps critique Tc_n}. Ensuite ce dernier fait des hypothèses sur la clé secrète K_h et calcule des valeurs internes relatives à ces hypothèses de clé. Dans le cas de la FSA présentée dans [Li 2012], la valeur interne considérée est le poids de Hamming de l'octet en entrée du bloc de SUBBYTE noté simplement dans cette section $HW_{n,h}(o)$ qui dépend de l'hypothèse de clé K_h et du message considéré M_n. L'attaquant obtient donc après cette seconde étape des triplets {Message M_n, Hypothèse de clé K_h, Poids de Hamming $HW_{n,h}$}. Finalement pour chaque hypothèse

de clé, K_h, les couples {Message (M_n), Temps critique Tc_n} et les couples {Message M_n, Poids de Hamming $HW_{n,h}$} sont corrélés en utilisant la corrélation de Pearson $\rho_{K_h}(Tc_n, HW_{n,h})$. La figure 1.11 illustre les résultats d'une FSA sur une implémentation de l'AES pour 360 messages différents. Chaque sous-figure représente les coefficients de corrélation pour les 256 hypothèses de clé pour chacun des octets de la clé secrète. Dans cet exemple, chaque octet de la clé a été retrouvé.

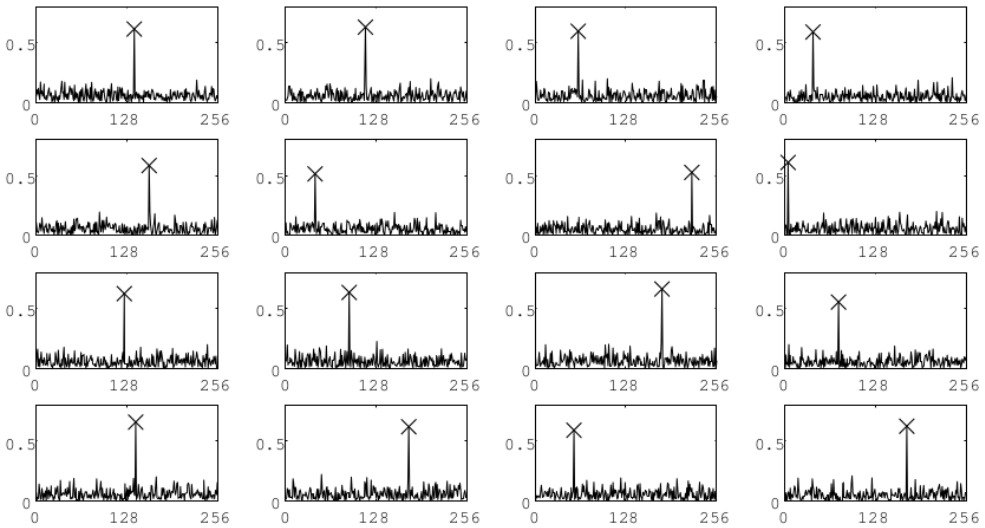

FIGURE 1.11 – Coefficient de corrélation $\rho_{K_h}(Tc_n, HW_{n,h})$ en fonction des hypothèses de clé K_h - extrait de [Li 2012]

1.3 Introduction aux injections de fautes par violation de contraintes temporelles

1.3.1 Contraintes temporelles des circuits synchrones

Une majorité des circuits numériques sont implémentés de façon synchrone, c'est-à-dire que leurs opérations sont cadencées par une horloge commune. Comme présenté dans la figure 1.12, un circuit synchrone est constitué d'un ou plusieurs blocs logiques (Σ) encadrés par des registres (DFF) cadencés par une horloge commune (clk).

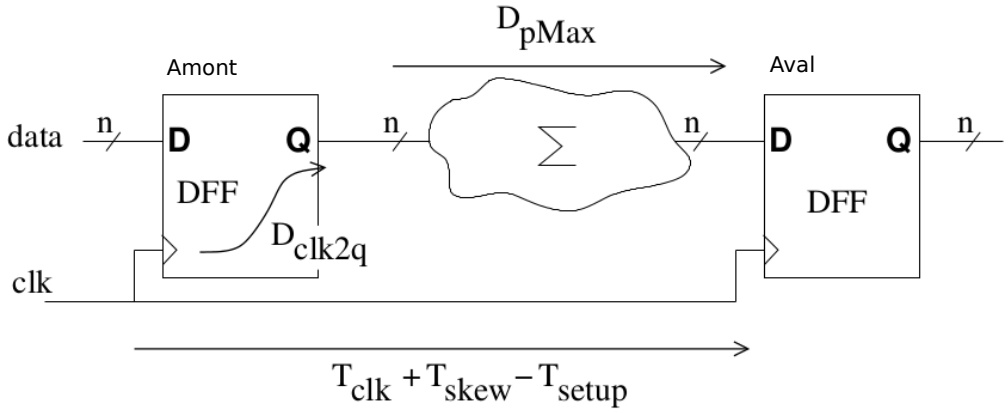

FIGURE 1.12 – Architecture de principe des circuits synchrones

Sur un front montant de l'horloge, les données sont mises à jour à la sortie des registres. Ensuite ces nouvelles données sont traitées par le bloc logique avant d'être une nouvelle fois enregistrées dans les registres au front montant d'horloge suivant. Aussi, il apparaît évident que la période d'horloge (T_{clk}) doit être plus longue que le temps le plus long de propagation des données dans la logique (D_{pMax}) pour assurer un bon déroulement des calculs. Il s'agit donc d'écrire de façon précise les contraintes temporelles que le circuit synchrone doit vérifier.

- D_{clk2q} est le délai après lequel les données sont effectivement mises à jour en sortie du registre après un front montant d'horloge.
- T_{skew} est la petite différence de phase qui peut exister entre deux signaux d'horloges cadençant les registres. Cette petite différence est due aux temps de propagation dans l'arbre d'horloge.
- T_{setup} est le temps pendant lequel la donnée doit rester stable en entrée d'un registre avant le coup d'horloge pour être échantillonnée correctement.
- T_{hold} est le temps pendant lequel la donnée doit rester stable en entrée d'un registre après le coup d'horloge pour être échantillonnée correctement.
- D_{pMax} est le temps maximal après lequel a lieu le dernier changement de valeur des données en entrée des registres avant d'être stable.
- D_{pMin} est le temps auquel a lieu le premier changement de valeur des données en entrée des registres après la mise à jour des données en entrée de la logique.

D'une part, l'équation 1.23 représente le temps de propagation le plus long.

$$\text{Temps de propagation maximal des données} : D_{clk2q} + D_{pMax} \tag{1.23}$$

D'autre part, l'équation 1.24 représente le temps maximal qu'ont les données pour

être stables en entrée des registres.

$$\text{Temps d'arrivée requis} : T_{clk} - T_{setup} + T_{skew} \tag{1.24}$$

En prenant en compte les deux équations précédentes l'équation 1.25 relative aux contraintes de temps de maintient (de setup), est obtenue :

$$T_{clk} > D_{clk2q} + D_{pMax} + T_{setup} - T_{skew} \tag{1.25}$$

Il existe une seconde contrainte temporelle : il s'agit de la contrainte sur le temps de hold. L'équation 1.26 illustre cette contrainte :

$$D_{clk2q} + D_{pMin} > T_{hold} \tag{1.26}$$

La figure 1.13 représente l'évolution d'un bit dans le cas où les contraintes temporelles sont respectées. On note que l'entrée du registre final (D_{aval}) subit de nombreux glitchs de logique avant de se stabiliser à sa valeur finale. Il existe une marge temporelle (slack) entre la dernière transition de ce signal et le début du temps de setup.

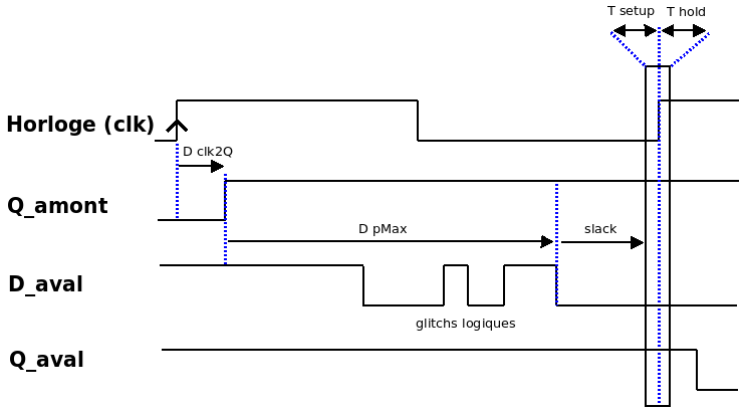

FIGURE 1.13 – Contraintes temporelles respectées

La violation de ces contraintes temporelles est une façon assez fréquente d'injecter des fautes dans un circuit. Deux étapes de ce mécanisme d'injection de fautes sont représentées dans les figures 1.14 et 1.15. Dans ces figures, la période d'horloge est diminuée progressivement jusqu'à ce que des violations de contraintes temporelles apparaissent.

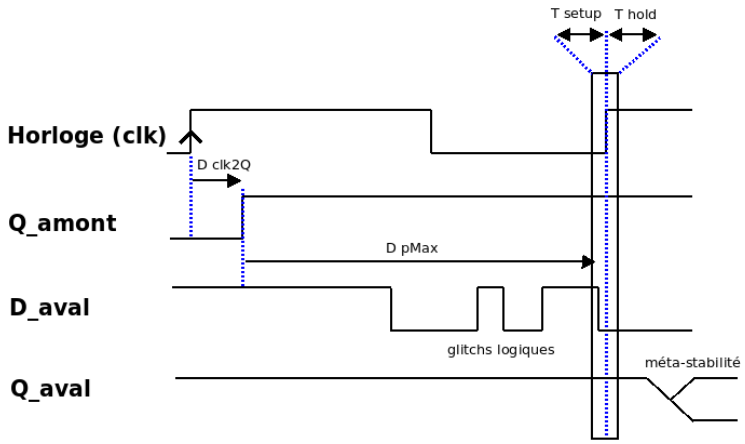

FIGURE 1.14 – Violation de la contrainte sur le temps de setup : état métastable

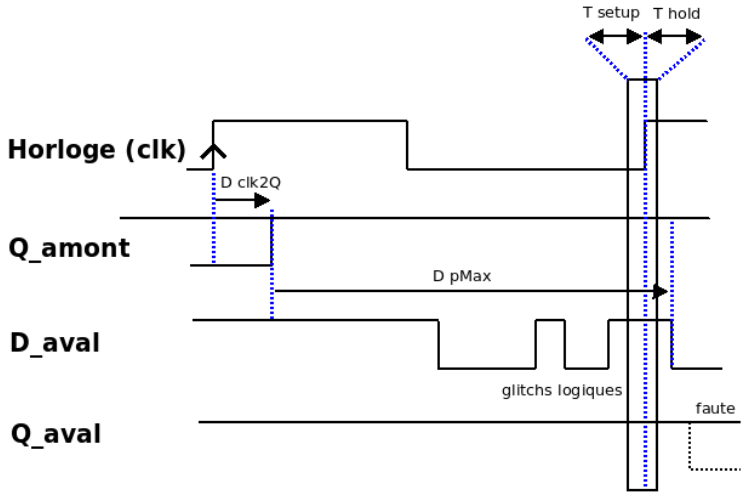

FIGURE 1.15 – Échantillonnage prématuré dû à une augmentation de la fréquence de fonctionnement du circuit

La zone encadrée autour du front montant d'horloge correspond à l'intervalle de temps pendant laquelle un changement de valeur en entrée du registre entraîne un comportement non déterministe de la sortie du registre, la métastabilité. Le phénomène de métastabilité est décrit dans [Stephenson 2009] et [Horstmann 1989]. Cet intervalle correspond au temps de setup avant le coup d'horloge et au temps de hold après le coup d'horloge. En cas de mauvais échantillonnage une faute est injectée.

La figure 1.14 présente le fonctionnement du circuit quand la dernière transi-

tion du signal d'entrée se trouve trop proche temporellement du front montant de l'horloge. Dans ce cas le registre se trouve dans un état métastable : sa sortie peut se stabiliser aussi bien à un état haut qu'à un état bas quelle que soit la valeur du signal en entrée. Une faute peut être injectée, ou non. La figure 1.15 introduit un autre comportement qui correspond à un échantillonnage prématuré. Dans ce cas une valeur erronée est échantillonnée par le registre et une faute est injectée de façon déterministe puisqu'il n'y a pas de transition du signal d'entrée dans la zone encadrée.

1.3.2 Violation de contraintes temporelles sur le temps de setup

1.3.2.1 Diminution de la période d'horloge

La première méthode, la plus évidente, pour induire une violation de temps de setup est d'augmenter la fréquence de fonctionnement du circuit (diminuer sa période). Dans ce cas, le mécanisme d'injection de fautes est sans ambiguïté dû à des violations de contraintes temporelles sur le temps de setup.

Cette technique n'est plus efficace sur les circuits sécurisés mais dans cette thèse elle permettra de construire des bibliothèques de référence qui seront ensuite utilisées comme base de comparaison pour les autres moyens d'injection de fautes étudiés.

Cette méthode est illustrée figure 1.15. Dans ce cas, la période d'horloge a été diminuée jusqu'à ce qu'une mauvaise valeur sur D_aval soit échantillonnée sur Q_aval au front montant de l'horloge (clk).

1.3.2.2 Augmentation des temps de propagation

La seconde façon d'injecter des fautes par violation de contraintes sur le temps de setup est d'augmenter les temps de propagations des données dans le circuit.

La figure 1.16 illustre un fonctionnement de circuit synchrone quand les temps de propagation ont été allongés de façon à ce que les contraintes temporelles ne soient plus respectées. Dans cette figure les temps de propagation ont augmentés, ce qui a pour effet de modifier le chronogramme D_aval jusqu'à ce qu'une mauvaise valeur soit échantillonnée sur Q_aval sur le front montant de l'horloge (clk).

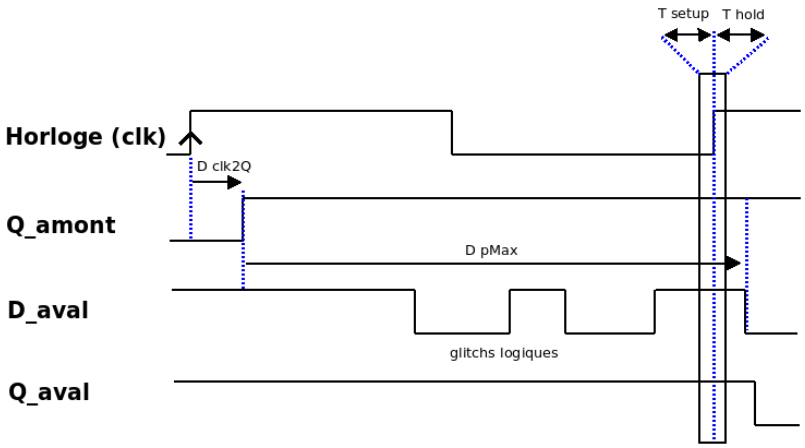

FIGURE 1.16 – Échantillonnage prématuré dû à une augmentation des temps de propagation dans le circuit

Pour identifier les paramètres pouvant influer sur le temps de propagation dans un circuit CMOS, l'équation des temps de propagation d'un inverseur est rappelée. Les équations sont évidemment plus complexes pour des chemins de données plus compliqués. Cependant, les tendances sont similaires. Le schéma de principe d'un inverseur en technologie CMOS est illustré par la figure 1.17. t_{pLH} et t_{pHL} représentent les temps de propagation pour un changement d'état de la sortie de l'inverseur passant respectivement d'un état bas à un état haut et d'un état haut à un état bas.

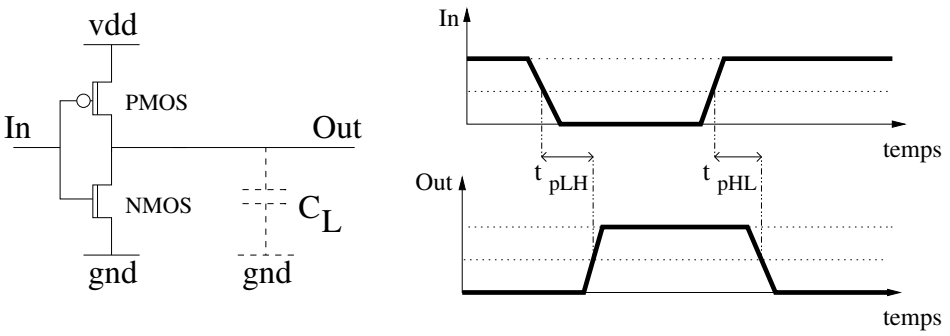

FIGURE 1.17 – Schéma de principe d'un inverseur

Le temps de propagation, t_{pLH} (équation 1.27), est obtenu avec une analyse de

premier ordre [Razavi 2008] du comportement dynamique d'un inverseur :

$$t_{pLH} = \frac{C_L \left[\frac{2|V_{th,p}|}{V_{DD} - |V_{th,p}|} + \ln\left(3 - 4\frac{|V_{th,p}|}{V_{DD}}\right)\right]}{\mu_p C_{ox} \frac{W_p}{L_p}(V_{DD} - |V_{th,p}|)} \tag{1.27}$$

Avec V_{DD} la tension d'alimentation, C_L la capacité de charge, $V_{th,p}$ la tension de seuil des transistors PMOS, μ_p la mobilité des trous, C_{ox} la capacité d'oxyde de grille et (W_p/L_p) le rapport largeur/longueur du transistor PMOS. Une équation similaire pour t_{pHL} peut être dérivée de l'équation 1.27 en remplaçant les paramètres liés au transistor PMOS par les paramètres liés au transistor NMOS (e.g. μ_n, (W_n/L_n), $V_{th,n}$).

Diminution de la tension d'alimentation

A partir de l'équation 1.27 nous constatons que si la tension d'alimentation V_{DD} diminue alors les temps de propagation d'un inverseur augmentent. Par extension, les temps de propagation au travers de blocs logiques en général seront augmentés avec la diminution de la tension d'alimentation. En pratique les temps de propagation augmentent de façon quasi-linéaire. De ce fait, sous-alimenter un circuit est un moyen d'injection de fautes par violation de contraintes temporelles.

Augmentation de la température

Dans l'équation 1.27, la mobilité des porteurs de charge et la tension de seuil des transistors sont les deux paramètres qui varient en fonction de la température. Cependant, les variations des temps de propagation relatives à V_{th} sont négligeables par rapport aux variations relatives à la mobilité des porteurs de charges μ. De ce fait, au premier ordre, seule la dépendance à la température de la mobilité des porteurs de charge μ est considérée [Ha 2011].

$$\mu(T) = \mu(T_0)\left(\frac{T}{T_0}\right)^{\alpha} \tag{1.28}$$

Avec T, la température et T_0 et α des paramètres de mise en forme (fitting). α varie approximativement de -2.2 à -1.5 en fonction du niveau de dopage [Ha 2011]. Alors en considérant les équations 1.27 et 1.28, les temps de propagation dans un circuit augmentent avec la température.

1.3.2.3 Variations transitoires de la tension

L'utilisation de variations dynamiques de tension (glitchs de tension) a été décrite dans la bibliographie [BarEl 2006, Cho 2005, Tummeltshammer 2009, Barenghi 2012] mais très peu d'articles ont véritablement traité le mécanisme d'injection de fautes relatifs à ces glitchs. Dans [Djellid-Ouar 2006] les auteurs montrent, sur la base de simulations, que les variations transitoires de tension ne peuvent pas induire de fautes dans un registre. De plus, l'article explicite que les fautes induites par une diminution de la tension sont liées à des violations de contraintes temporelles. Cela dit, d'autres explications du mécanisme d'injection de fautes lié aux glitchs de tension peuvent être trouvées dans la bibliographie. Les auteurs de [Yanci 2009] suggèrent que les glitchs de tensions induisent des différences de niveau de tension entre les sous-parties d'un circuit et que ce sont ces différences qui sont à l'origine de l'injection de fautes.

1.3.2.4 Impulsions électromagnétiques

Au début des années 2000, les variations transitoires de l'environnement électromagnétique (glitchs électromagnétiques ou impulsions électromagnétiques) ont été introduites par Quisquater *et al.* [Quisquater 2002]. Elles sont devenues par la suite un moyen d'injection de fautes de plus en plus rependu [Schmidt 2007, Dehbaoui 2012b, Dehbaoui 2013, Bayon 2012]. Ces travaux indiquent que le mécanisme d'injection de fautes peut être dans certains cas lié à des violations de contraintes temporelles. Des violations de contraintes temporelles seraient induites par des diminutions transitoires de la tension d'alimentation du circuit liées aux perturbations de l'environnement électromagnétiques. Ces effets dépendent en grande partie de la qualité du couplage électromagnétique qui existe entre le réseau d'alimentation de la cible et les perturbations électromagnétiques. De plus, ces diminutions de tension induites par les perturbations électromagnétiques semblent avoir un effet local.

1.3.3 Propriétés de l'injection par violation des contraintes temporelles

Les fautes induites par des violations de contraintes temporelles se caractérisent de la façon suivante :

– Les fautes injectées sont dépendantes des données traitées. Si les données en

entrée changent, alors les temps de propagation de chaque ronde sont différents et donc la perturbation qu'il faudra appliquer pour injecter une faute sera différente.

– Un phénomène de métastabilité apparaît (comme illustré dans la figure 1.14), dans un premier temps. Si la perturbation devient plus forte la sortie redeviendra alors déterministe (comme illustré dans la figure 1.15).

Comme présenté dans la figure 1.14 si les données ne sont pas stables assez tôt avant le coup d'horloge (ou assez longtemps après), le registre concerné entre alors dans un état métastable. Dans cet état la sortie du registre peut converger de façon théoriquement équiprobable vers un état haut ou vers un état bas après un temps indéterminé. Cependant cet état métastable est difficilement observable puisque la gigue d'horloge (instabilité de la période d'horloge autour de sa valeur nominale) doit aussi être prise en compte dans les équations de contraintes temporelles. Ces deux phénomènes peuvent être à l'origine du non-déterminisme d'un échantillonnage.

La figure 1.18 présente le taux d'occurrence de fautes injectées en fonction de la fréquence de fonctionnement imposée au circuit. La plage de non-déterminisme (taux d'injection différent de 0% ou 100%) peut être observée. Ces acquisitions ont été faites sur les outils qui seront présentés dans le chapitre 2.

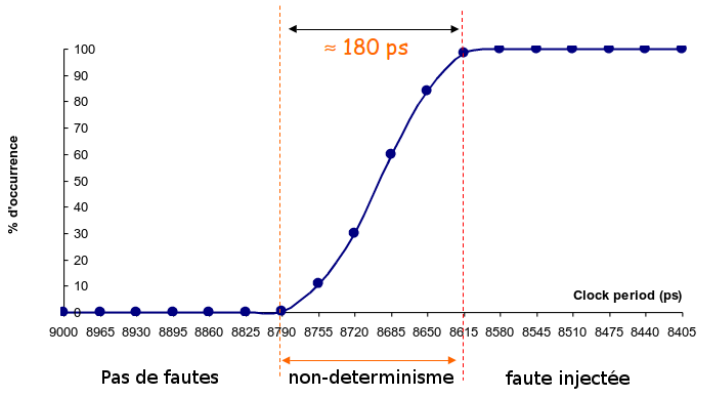

FIGURE 1.18 – Taux d'occurrence de fautes par violation de contraintes temporelles sur le temps de setup en fonction de la fréquence

La figure 1.19 est une acquisition faite à l'oscilloscope illustrant la gigue d'horloge du générateur de fréquence utilisé dans cette thèse. Sur cette figure le signal d'horloge délivré par le générateur est observé en mode persistance ce qui permet de distinguer graphiquement la plage de variation d'une période d'horloge. La période nominale

de cette horloge est de 10ns et la gigue d'environ 175ps.

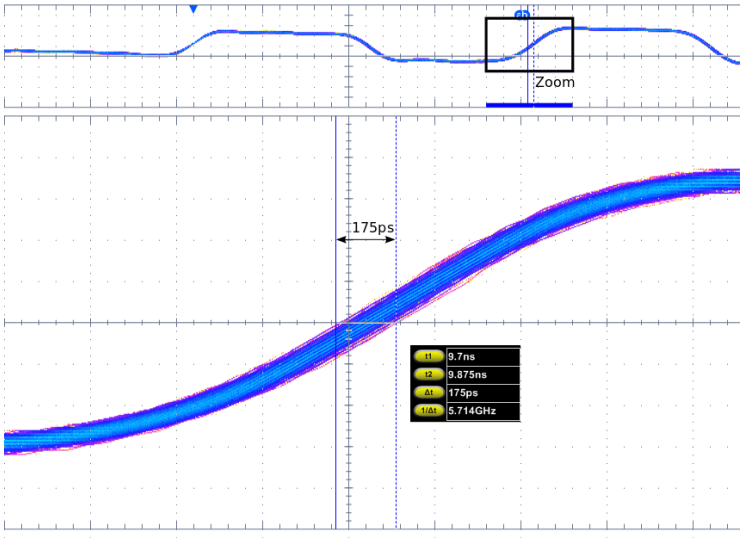

FIGURE 1.19 – Acquisition à l'oscilloscope de la gigue d'horloge du générateur de fréquence

1.4 Conclusion

Dans un premier temps, différents algorithmes de chiffrement ont été présentés avec un focus sur l'AES car une implémentation matérielle de ce standard de chiffrement servira de cible dans la suite de ce travail.

Ensuite, un état de l'art des techniques de cryptanalyse physique passives (basées sur l'observation du comportement du circuit) ou actives (basées sur l'injection de fautes) a été proposé. Quelques méthodes de cryptanalyse active ont été explicitées pour mettre en avant l'importance d'une bonne maîtrise du mécanisme d'injection de fautes.

Dans le cadre de cette thèse, nous avons fait le choix de nous concentrer sur l'étude des injections de fautes par violation de contraintes temporelles relatives aux temps de setup. Ces contraintes et les méthodes qui seront mises en œuvre pour les fauter ont donc été détaillées.

Dans la suite de ce document, les bancs d'injection pouvant entraîner des violations de contraintes temporelles ainsi que des outils d'analyse permettant d'étudier ce mécanisme d'injection seront décrits.

Chapitre 2

Description des outils et bancs d'analyse

Sommaire

Dans ce chapitre, les outils et les bancs d'analyse utilisés dans le cadre de cette thèse sont présentés. Il s'agit d'une part de la cible matérielle sur laquelle sera implémenté l'algorithme de chiffrement AES et d'autre part des bancs d'injection de fautes. Une implémentation matérielle d'un voltmètre intégré est aussi décrite ainsi qu'un détecteur d'injection dont l'efficacité sera étudiée dans les chapitres 4 et 5.

2.1 Implémentation matérielle de l'AES-128 sur FPGA

L'AES a été utilisé comme cible dans le cadre de cette thèse car il s'agit d'une cible classique lors d'attaques sur circuits sécurisés.

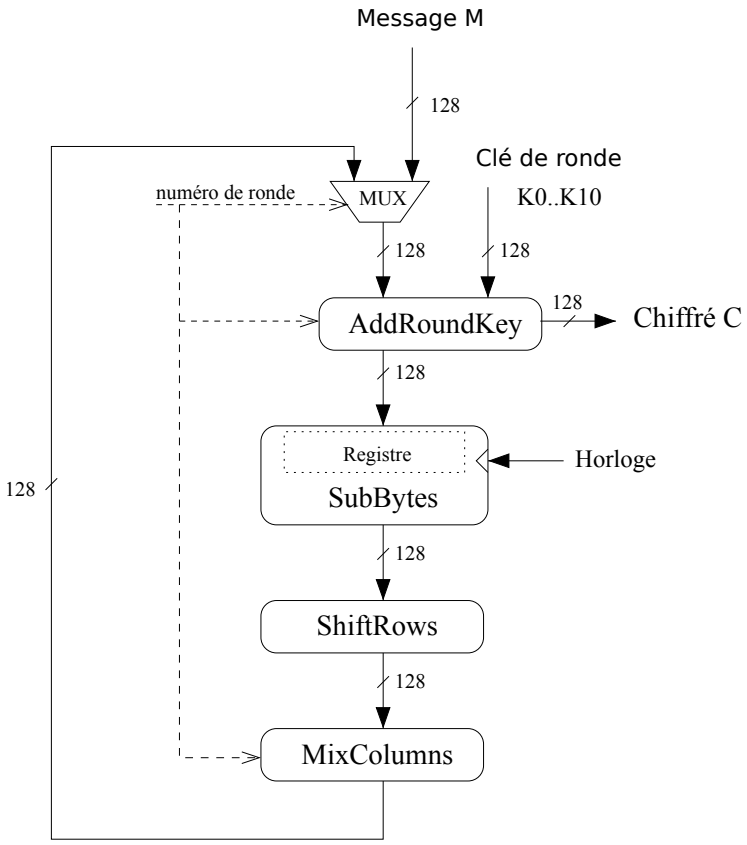

FIGURE 2.1 – Implémentation de l'algorithme de chiffrement AES-128

Une version de l'AES-128 a été implémentée avec une architecture en boucle sur un FPGA (Xilinx Spartan 3A) comme présentée dans la figure 2.1. Un bloc de communication permet la communication RS-232 entre le FPGA et l'ordinateur. Une machine d'état traite les commandes envoyées par l'ordinateur (modification de la clé secrète, modification du message à chiffrer, exécution du chiffrement et retour du message chiffré, etc.). Le bloc KEY EXPANDER gère le calcul "à la volé" des différentes clés de ronde (K1 à K10). Ce bloc qui n'est pas présenté sur la figure 2.1 est décrit dans la section 1.1.4. L'implémentation de la cible sur FPGA permet de reconfigurer facilement le matériel pour l'adapter aux besoins spécifiques d'une attaque. Elle permet aussi l'ajout de contre-mesures dans un second temps.

Matériellement, les S-Box sont implémentées au moyen de tables d'allocation (look-up table, LUT).

Un signal de synchronisation est envoyé sur une sortie du FPGA pour faciliter la synchronisation des bancs d'attaque. Il a été conçu pour être envoyé 30 ou 300ns

avant le début du chiffrement.

Chaque ronde de l'AES s'exécute en une période d'horloge. Un chiffrement complet se fait donc en 11 périodes d'horloge. Le système fonctionne avec une horloge de 100MHz. Cette horloge peut être fournie soit par un quartz 50MHz présent sur la maquette et remise en forme avec la PLL du FPGA soit elle peut être directement récupérée via le port SMA de la maquette.

2.2 Bancs d'injection de fautes

2.2.1 Banc d'injection de fautes par modification de la fréquence

Les perturbations de l'horloge interne d'un circuit peuvent se faire de façon "statique", c'est-à-dire que l'augmentation de la fréquence de fonctionnement durera tout au long du fonctionnement du circuit : "overclocking". Ces perturbations peuvent aussi être transitoires et n'affecter qu'un cycle d'horloge : glitchs d'horloge. Aujourd'hui les circuits sécurisés sont relativement bien protégés contre ce genre d'attaque (horloge interne remise en forme à l'aide de PLL par exemple). Cependant, même si les glitchs d'horloge ne peuvent plus être considérés comme une technique d'attaque actuelle, ils demeurent utiles pour la caractérisation et l'analyse des vulnérabilités des circuits.

Les modifications statiques de la fréquence ont été effectuées à l'aide d'un générateur de d'horloge (Holzworth HS3001A) pilotable par USB.

Les modifications dynamiques de la fréquence ont été effectuées à l'aide d'un FPGA sur lequel un générateur de glitchs d'horloge a été implémenté comme introduit par [Dutertre 2009, Fukunaga 2009, Agoyan 2010b, Endo 2011]. Cette technique permet de choisir précisément la perturbation appliquée au circuit cible (la réduction de la période d'horloge) par pas de 35ps et de ce fait, de contrôler le nombre de fautes injectées. Il est aussi possible de choisir précisément la ronde de l'AES ciblée, ce qui est une propriété très importante pour mener à bien une attaque de type DFA (voir section 1.2.2.1).

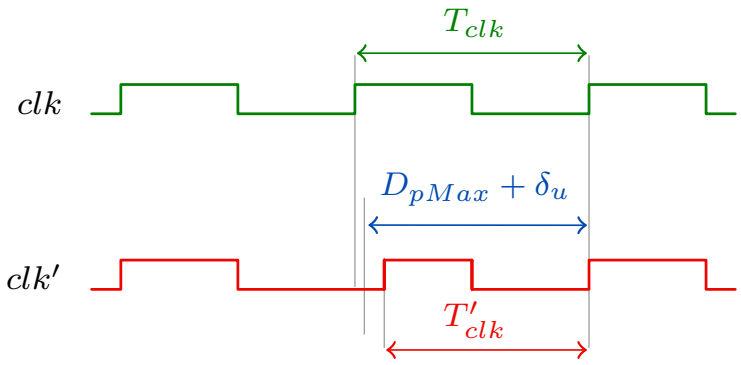

FIGURE 2.2 – Glitch d'horloge théoriquement injecté

Dans le cadre de cette thèse une méthode similaire a été mise en œuvre. Ce banc d'injection de glitch d'horloge est implémenté sur un FPGA Spartan Virtex 5. La figure 2.2 présente le glitch théorique qui est formé par le FPGA. Cette implémentation fournie une horloge de période nominale de 10ns (T_{clk}). Une période de ce signal d'horloge peut être réduite par pas de 35ps, (T'_{clk}). Quand la période d'horloge raccourcie devient trop courte, $T'_{clk} = T_{clk} - \Delta T < D_{pMax} + \delta_u + (D_{clk2q} - T_{skew})$, des violations de contraintes temporelles sont injectées. La synchronisation avec un signal de trig permet de viser la ronde du circuit cible désirée. D'autre part, la première trace visualisée sur la capture d'oscilloscope présentée figure 2.3 est le signal d'horloge non glitché, clk. Et la seconde trace représente le signal d'horloge glitché, clk', (dans le cas présenté sur la figure $\Delta T = 3{,}5$ns).

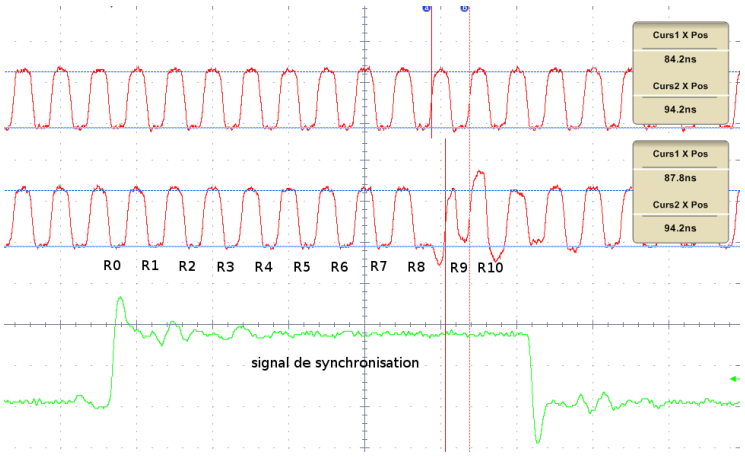

FIGURE 2.3 – Capture d'oscilloscope d'un glitch d'horloge

Enfin, la figure 2.4 présente le banc d'injection de glitchs d'horloge. Le FPGA de gauche (Virtex 5) génère l'horloge glitchée qui cadence le FPGA de droite (Spartan 3) sur lequel l'AES est implémenté. La synchronisation se fait avec le signal de synchronisation (fil rouge).

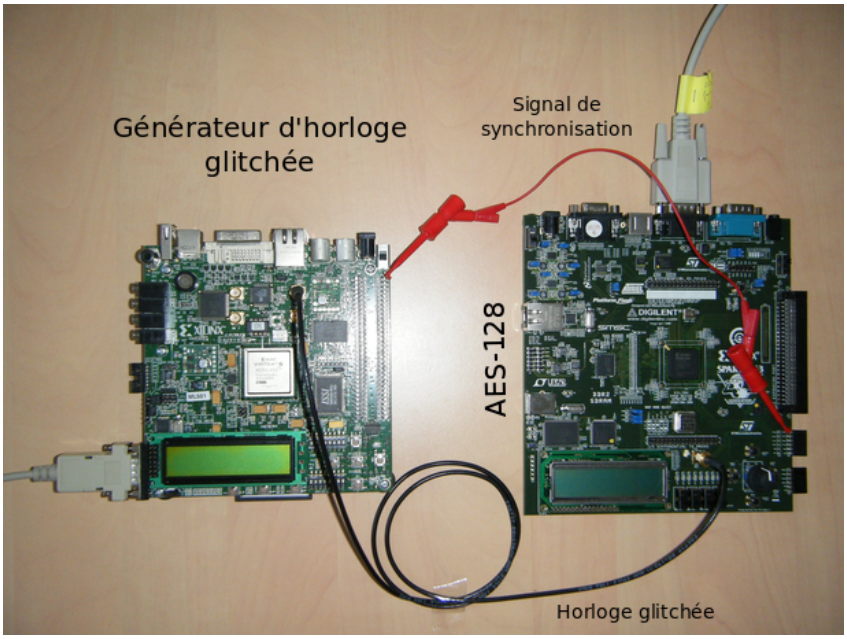

FIGURE 2.4 – Banc d'injection de glitchs d'horloge / Banc de mesure des temps critiques

2.2.2 Banc d'injection de fautes par modification de la température

L'augmentation de la température entraîne une augmentation des temps de propagation dans la logique implémentée sur silicium. L'étude de ce phénomène sera détaillée dans le chapitre 3. La figure 2.5 représente le banc de chauffe qui est constitué d'un souffleur d'air. La température est réglée manuellement à l'aide d'un potentiomètre. Une protection en alumine (blanche) confine la chaleur autour du FPGA. Le montage étant assez empirique et la température n'étant pas asservie de façon précise, un thermomètre a été positionné sur le FPGA pour mesurer au mieux la température effective en surface du boiter au moment de l'injection.

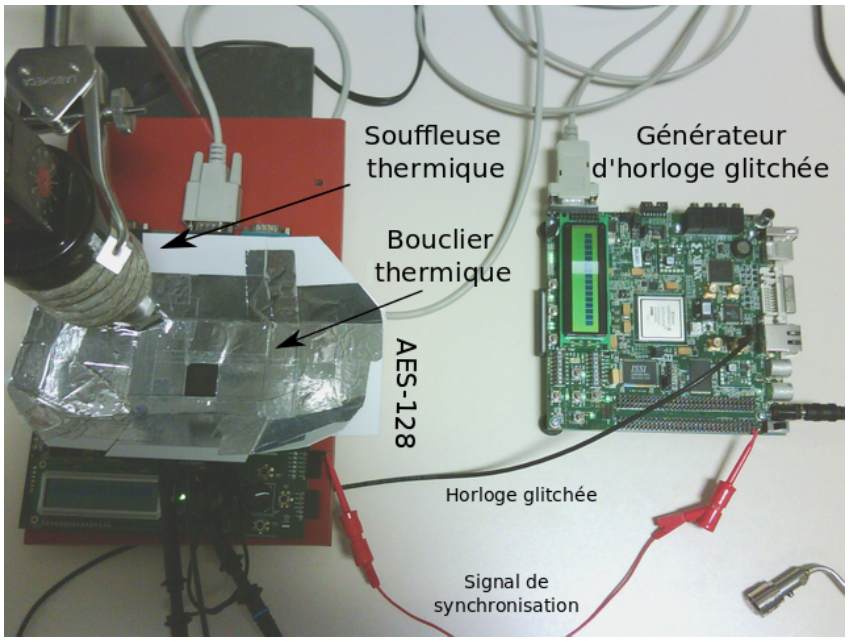

FIGURE 2.5 – Banc de chauffe

2.2.3 Banc d'injection de fautes par modification de la tension d'alimentation

La figure 2.6 représente le banc d'injection constitué de deux générateurs d'impulsions de tension directement branchés sur l'alimentation externe du FPGA (maquette retournée au premier plan). Les deux générateurs d'impulsion (Agilent 8114A et Picosecond 10,300B) permettent la génération de signaux rectangulaires dont l'amplitude, la durée, l'offset continu et le moment d'injection après un signal de synchronisation peuvent être réglés dans les plages indiquées par le tableau 2.1. Du fait qu'un offset statique puisse être ajouté, l'utilisation de T-bias n'est pas nécessaire. Pour des raisons pratiques, un autre générateur d'alimentation stabilisée (Hameg HMP2030) est utilisé pour les variations statiques de la tension.

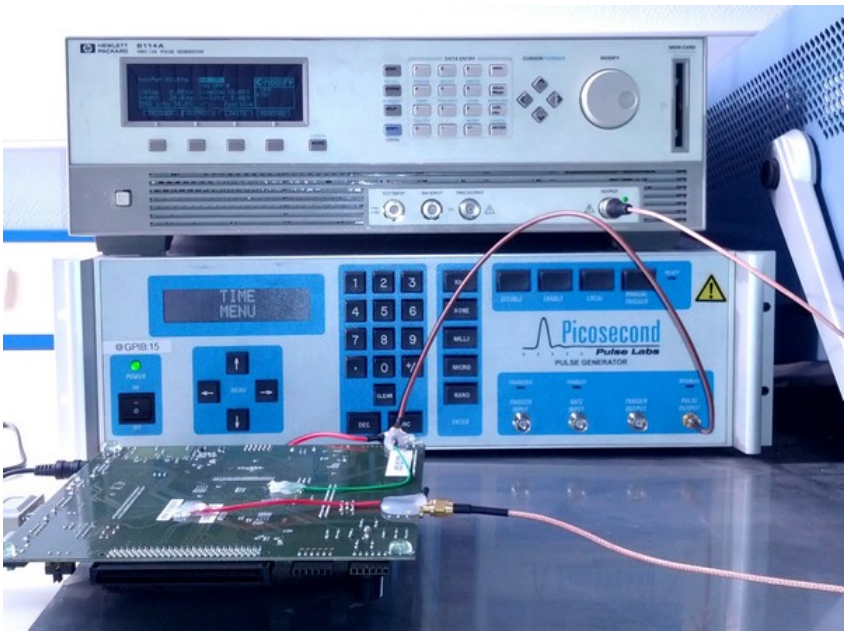

FIGURE 2.6 – Banc d'injection de glitchs de tension

Lors d'une attaque statique la tension d'alimentation est diminuée pas à pas jusqu'à l'apparition de fautes. Lors d'une attaque dynamique le même protocole est appliqué mais une variation transitoire de tension est superposée au signal continu.

TABLE 2.1 – Caractéristiques des

Ref.	Plage d'amplitude	Tps de monté	Tps de descente	Durée d'impulsion
Agilent 8114A	1V $\leftrightarrow$ 50V	10ns	10ns	10ns $\leftrightarrow$ 100ms
Picosecond 10,300B	-50V $\leftrightarrow$ +50V	$\leq$300ps	$\leq$300ps	1ns $\leftrightarrow$ 100ns

2.2.4 Banc d'injection de fautes par impulsions électromagnétiques

Le banc d'injection d'impulsions électromagnétiques est constitué d'une table motorisée contrôlable par ordinateur et d'un générateur d'impulsions de tension relié à une antenne qui servira d'injecteur. La forme du champ électromagnétique généré par une même impulsion peut être modifiée en changeant les paramètres de l'antenne (diamètre, nombre de spires, etc.) ou en changeant son orientation. L'antenne peut ensuite être déplacée au dessus de la cible à l'aide de la table XYZ contrôlable avec une précision de $10\mu m$.

Le générateur quant à lui est capable de générer des impulsions jusqu'à 200 V (positives ou négatives) avec un fort courant (jusqu'à 4A). Ces impulsions de tension sont envoyées dans l'antenne (50 Ω d'impédance) qui est constituée d'un fil de cuivre enroulé autour d'un noyau en ferrite. Quand le courant passe au travers de la bobine de cuivre, un champ magnétique est généré. Il est ensuite concentré par le noyau en ferrite de l'antenne. La durée de l'impulsion peut varier de 10ns à 200ns. Les temps de montée et de descente des impulsions sont fixées à 2ns.

La figure 2.7 représente le schéma du fonctionnement du banc d'injection de glitchs électromagnétiques qui a été utilisé dans notre étude.

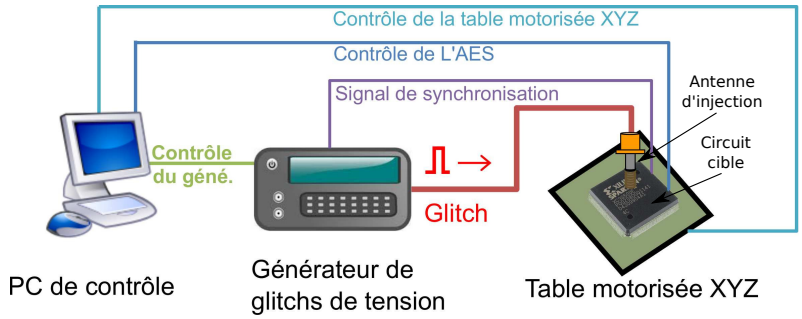

FIGURE 2.7 – Schéma de principe du banc d'injection de glitchs électromagnétiques

La figure 2.8 illustre à titre d'exemple plus spécifiquement une antenne propriétaire et la circuit cible.

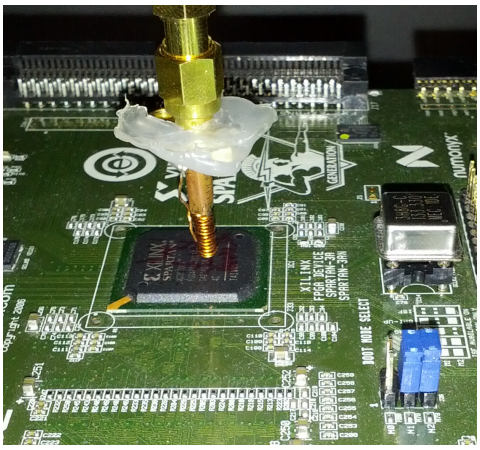

FIGURE 2.8 – Banc d'injection de glitchs électromagnétiques

2.3 Voltmètre embarqué

Afin d'observer l'effet réel des injections de glitchs de tension sur la tension interne du FPGA, un voltmètre intégré a été implémenté. Ce voltmètre permet de construire une image de la tension interne du circuit plus précise que celle observée avec un oscilloscope externe.

2.3.1 Principe

Mesurer des temps de propagation revient à mesurer la tension interne du circuit. Ainsi, dans l'optique de mesurer cette tension interne quand le circuit est exposé à des glitchs de tension, un délai-mètre (voltmètre basé sur la mesure des temps de propagation) a été conçu. Nous nous sommes appuyé sur une technique déjà introduite sur FPGA par K. Zick et al. [Zick 2013].

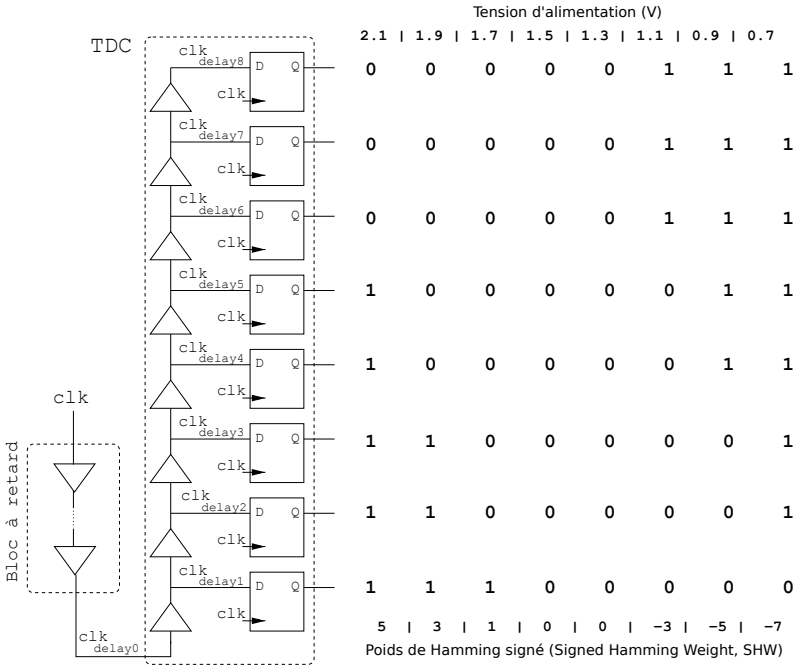

FIGURE 2.9 – Schéma de principe du fonctionnement du voltmètre intégré

La figure 2.9 représente l'architecture simplifiée du délai-mètre. Il est constitué de deux blocs :

- Un bloc à retard dont le délai dépend de la tension interne,
- Un convertisseur "time-to-digital" (TDC [Dudek 2000]) utilisé pour transformer une différence de phase entre deux signaux en un code binaire.

L'entrée du délai-mètre est le signal d'horloge du circuit, clk. Ce signal est retardé à travers le bloc à retard. La nouvelle horloge, $clk_{delay(0)}$, est retardée d'un retard, $delay(Vdd)$, qui dépend de la tension d'alimentation Vdd. Ensuite, le TDC est utilisé pour convertir la différence de phase entre l'horloge clk et l'horloge retardée $clk_{delay(0)}$ en un code binaire. Cette différence de phase (égale à $delay(Vdd)$) dépend linéairement de Vdd. Le TDC est constitué d'une série de 8 éléments retardant le signal d'un faible retard, δd. En entrée de ce TDC, $clk_{delay(0)}$ est injectée créant ainsi 8 sous-horloges avec des petits retards additionnels : $n * \delta d$ avec n le nombre d'éléments retardants traversés par l'horloge. Ensuite, 8 registres échantillonnent les horloges retardées sur le front montant de l'horloge principale, clk. La n^{ime} sortie est à l'état bas si $clk_{delay(n)}$ est en avance de phase par rapport à l'horloge principale. Réciproquement, elle est à l'état haut si $clk_{delay(n)}$ est en retard de phase. Finalement les n sorties des DFFs forment un vecteur de 8 bits qui dépend du retard, $delay(Vdd)$, induit dans le bloc retardant ([Zick 2013, Dudek 2000]). La partie droite de la figure 2.9 représente le vecteur de sortie obtenu pour différentes valeurs de tension d'alimentation. Quand Vdd varie, la différence de phase entre l'horloge retardée et l'horloge principale varie et donc le vecteur de sortie varie aussi.

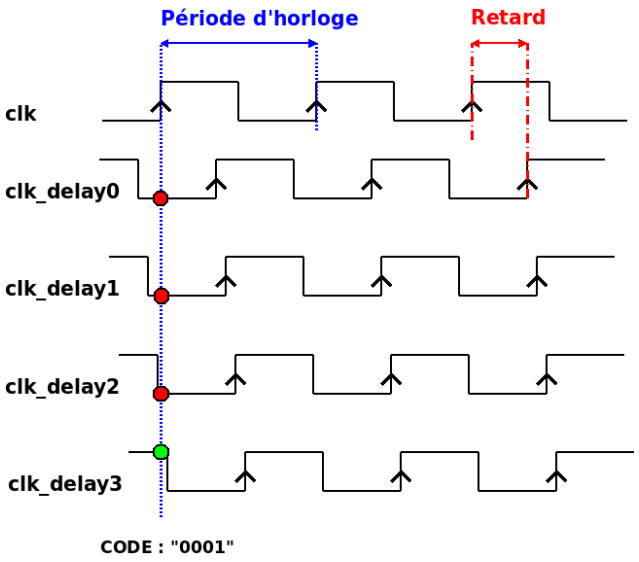

FIGURE 2.10 – Chronogramme d'un convertisseur (TDC) pour une tension de 0.9V

Les figures 2.10, 2.11 et 2.12 représentent les chronogrammes relatifs au principe de fonctionnement du convertisseur TDC pour différentes tensions. En fonction de la tension d'alimentation le délai principal, noté *Retard* sur les figures, augmente à

mesure que Vdd diminue. En fonction de cette valeur de délai, les valeurs échantillonnées sur le front montant de l'horloge principale sont différentes.

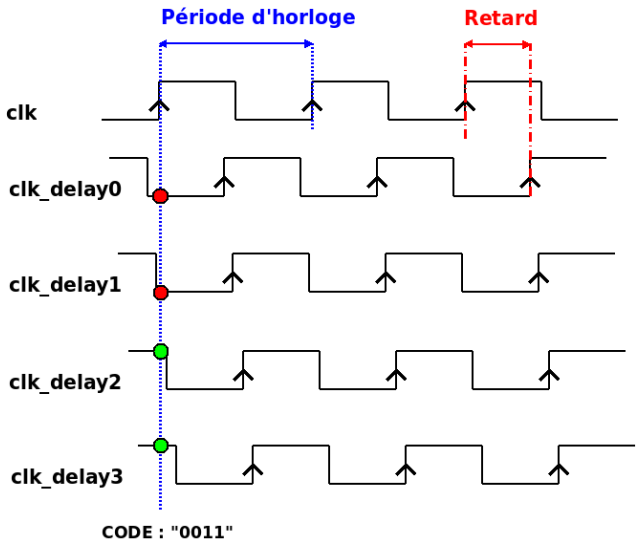

FIGURE 2.11 – Chronogramme d'un convertisseur (TDC) pour une tension de 0.8V

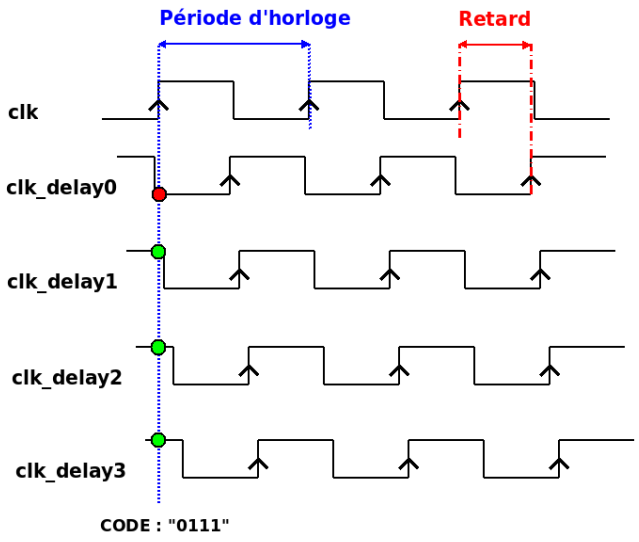

FIGURE 2.12 – Chronogramme d'un convertisseur (TDC) pour une tension de 0.7V

Le passage progressif de la tension de 0,9V à 0,8V puis à 0,7V, s'accompagne d'une évolution du code binaire obtenu : 0001, 0011 puis 0111.

Ce code binaire obtenu est un code dit "thermomètre" : il est formé de deux

blocs consécutifs de "0" et de "1". Les deux informations contenues dans ce code sont :

- Son poids de Hamming (HW),
- L'ordre de ces deux blocs, pour différencier "000111" et "111000" par exemple.

Pour prendre en compte ces deux informations le poids de Hamming signé (*Signed* Hamming Weight, SHW) sera considéré. Par exemple "00000111" et "11100000" seront respectivement notés "+3" et "-3" (voir la figure 2.9).

2.3.2 Implémentation

Comme l'illustre la figure 2.9, il existe une plage de tension pour laquelle le TDC renvoie un vecteur nul (de 1,5V à 1,3V sur la figure). Cette plage de tension est appelée "zone aveugle". Pour éliminer cette zone aveugle il existe plusieurs solutions :

- L'agrandissement du TDC, c'est à dire l'ajout de petits éléments retardants pour passer de 8 à 32 par exemple.
- L'implémentation de plusieurs délai-mètres dont les zones aveugles sont différentes, c'est cette solution qui a été retenue.

Quatre instances du délai-mètre précédemment présenté ont donc été implémentées sur le FPGA cible. Afin que les 4 implémentations soient identiques, une *hardmacro* du délai-mètre a été conçue et instanciée à 4 positions différentes sur le FPGA. Une *hardmacro* est un bloc logique (une *macro*) enregistré après l'étape de placement-routage. Ces étapes de placement et routage peuvent être faites avec précision à la main pour avoir un contrôle totalement maîtrisé sur les caractéristiques du bloc logique. Ensuite une *hardmacro* peut être instanciée autant de fois que nécessaire sur le FPGA et ces instances seront toutes matériellement identiques. Du fait des variations de fabrication dans un même circuit, les 4 instances ont quand même des retards, $delay(Vdd)$, légèrement différents. Les figures 2.13 et 2.14 représentent les plages de fonctionnement de 2 instances de délai-mètre obtenues de façon expérimentale après implémentation. La zone aveugle du second délai-mètre est légèrement décalée par rapport à la celle du premier (environs 0,2V). Aussi quelle que soit la plage de tension considérée, il existe un délai-mètre parmi les 4 fonctionnant hors de sa zone aveugle.

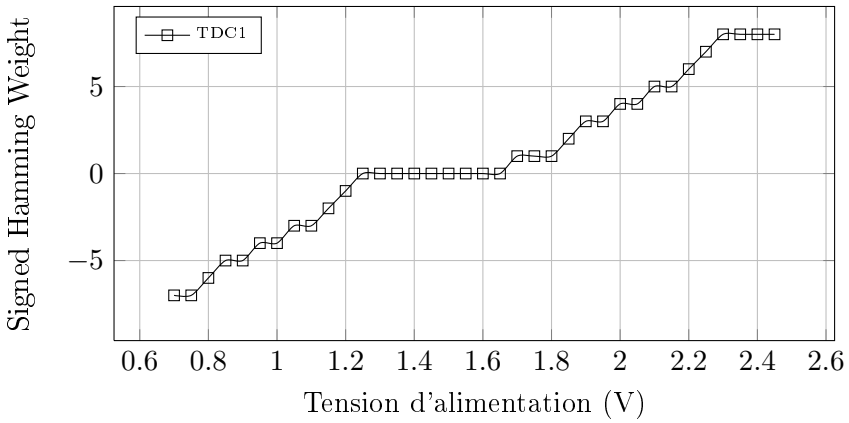

FIGURE 2.13 – Sortie du premier délai-mètre

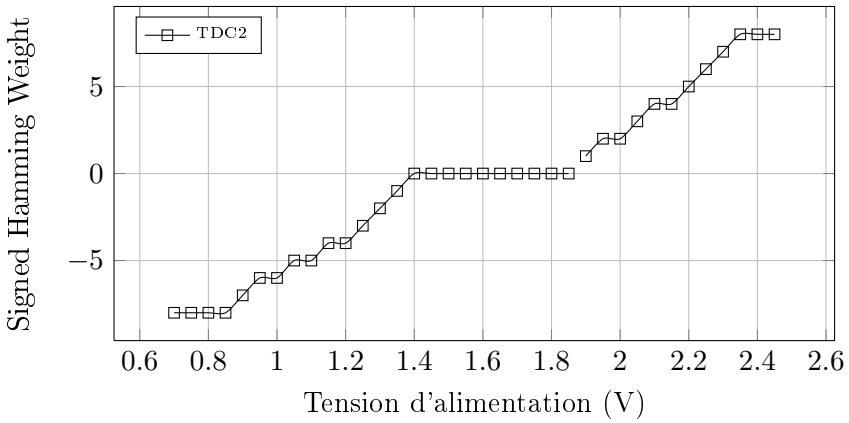

FIGURE 2.14 – Sortie du second délai-mètre

Finalement, en considérant la moyenne des 4 poids de Hamming signés, le voltmètre fonctionne sur une plage de tension allant de 0.7V à 2.4V avec une résolution d'environ 20mV (la tension nominale du FPGA est de 1.2V). Évidemment cette résolution dépend du nombre de délai-mètre fonctionnant hors de leur zone aveugle. De fait, cette résolution n'est pas constante. La figure 2.15 représente la relation entre la moyenne des SHW et la tension interne du circuit. Ces valeurs ont été obtenues expérimentalement après implémentation du voltmètre sur le circuit. Il s'agit de mesures faites en statique, c'est à dire obtenues pour une tension gardée identique pendant toute l'acquisition. La même courbe de référence est ensuite utilisée en dynamique pour faire correspondre un code SHW à une valeur de tension. En

pratique, le code SHW est échantillonné dans un registre à décalage à une fréquence de 200MHz.

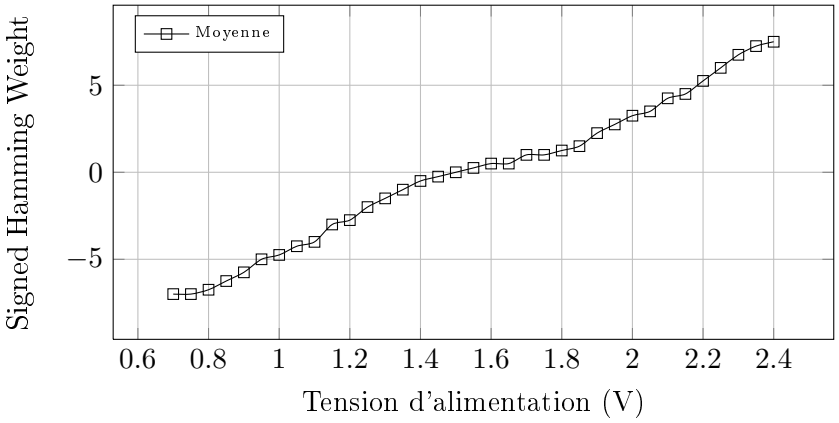

FIGURE 2.15 – Moyenne des sorties des 4 délai-mètres

2.4 Détecteur d'injection de fautes par violation de contraintes temporelles, DVCT

Dans [Endo 2012] et [Selmane 2011], les auteurs proposent un détecteur de violation de contraintes temporelles. Ce détecteur a été étudiée afin de vérifier son efficacité vis à vis d'injections électromagnétiques ainsi que les vulnérabilités qu'il induit.

Le principe de fonctionnement du détecteur est basé sur l'insertion d'un délai de garde (D_{garde}) tel qu'illustré dans la figure 2.16.

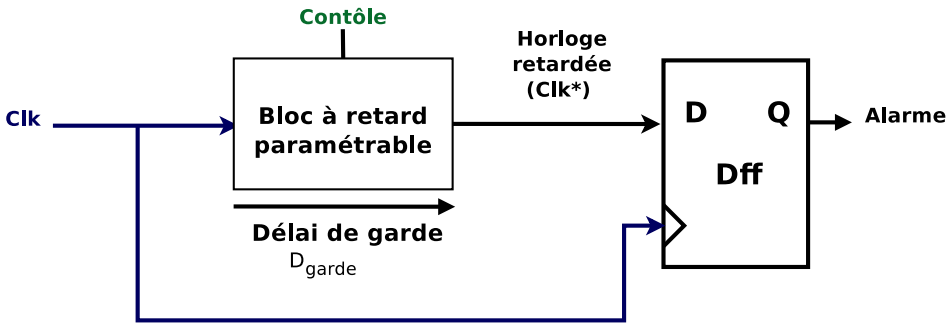

FIGURE 2.16 – Détecteur à délai de garde paramétrable

Ce délai de garde (D_{garde}) doit être compris entre le temps critique du circuit

(D_{pMax}) et sa période d'horloge (T_{clk}), de telle sorte qu'il soit plus long que les temps critiques de l'AES mais plus court que la période d'horloge. L'équation 2.1 et la figure 2.17 illustrent ces contraintes.

$$T_{clk} - T_{setup} + T_{skew} > D_{clk2q} + D_{garde} > D_{clk2q} + D_{pMax} \tag{2.1}$$

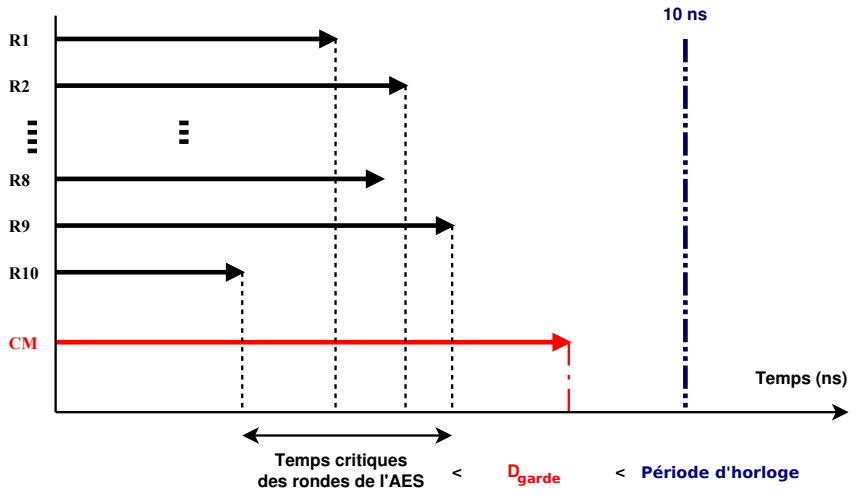

FIGURE 2.17 – Positionnement temporel du délai de garde

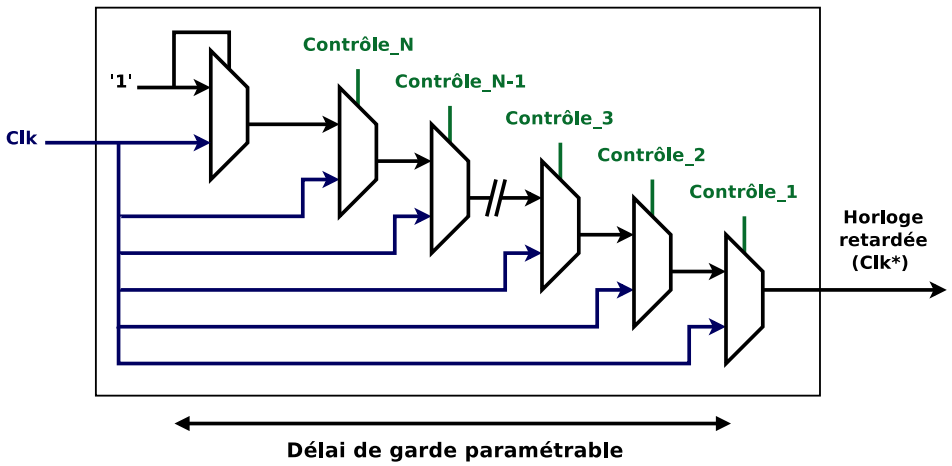

FIGURE 2.18 – Délai de garde paramétrable

Pour garantir que le délai de garde soit plus long que les chemins critiques de l'AES, les temps critiques de l'AES sont mesurés pour un grand nombre de couples {Message, Clé secrète} aléatoires. Ensuite le délai de garde du détecteur est

paramétré de façon à être supérieur aux temps critiques mesurés (auquel une marge est ajoutée : slack). De plus, une autre marge temporelle doit exister entre le délai de garde et la période d'horloge pour prévenir des faux positifs (détections inopportunes). La figure 2.18 illustre le fonctionnement du bloc à retard paramétrable. Ce bloc est constitué de multiplexeurs en cascade. Un signal de contrôle permet de définir le nombre de multiplexeurs que l'horloge (Clk) doit traverser pour former l'horloge retardée (Clk^*)

Ainsi toute violation des contraintes temporelles est précédée d'une violation du délai de garde qui déclenche une alarme. Ce détecteur a été conçu de façon à répondre aux contraintes suivantes :

- Le registre DFF doit échantillonner un '1' quand le circuit fonctionne normalement.
- Le registre DFF doit échantillonner un '0' quand le circuit subit une attaque susceptible d'induire une faute.

La figure 2.19 représente les chronogrammes de fonctionnement du détecteur soumis à une diminution dynamique de la fréquence. Quand l'horloge subit une modification d'une de ses périodes, Clk_{glitch}, cette modification se propage à travers le bloc de délai du détecteur, Clk^*_{glitch}. La différence de phase entre Clk_{glitch} et Clk^*_{glitch} entraîne l'échantillonnage d'un '0' et le déclenchement d'un signal de détection.

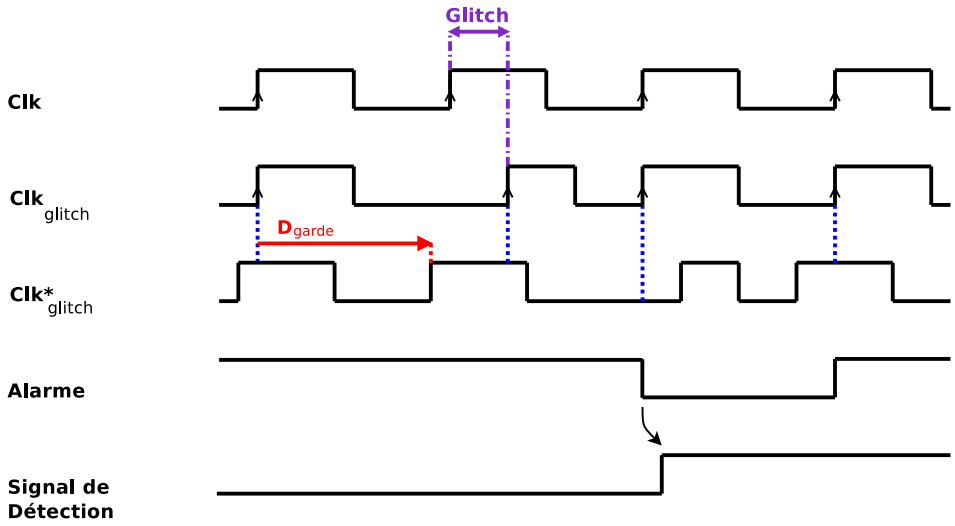

FIGURE 2.19 – Détecteur soumis à une injection de glitchs d'horloge

De plus, si le circuit est soumis à une attaque visant à induire des fautes basées

sur les violations de contraintes temporelles, le détecteur est aussi affecté. Son délai de garde étant plus grand que celui de l'AES, une faute sera injectée sur chemin critique du détecteur avant d'être injectée sur l'algorithme cryptographique. La figure 2.20 illustre le fonctionnement de ce détecteur soumis à une perturbation visant à augmenter les temps de propagation par diminution de la tension d'alimentation.

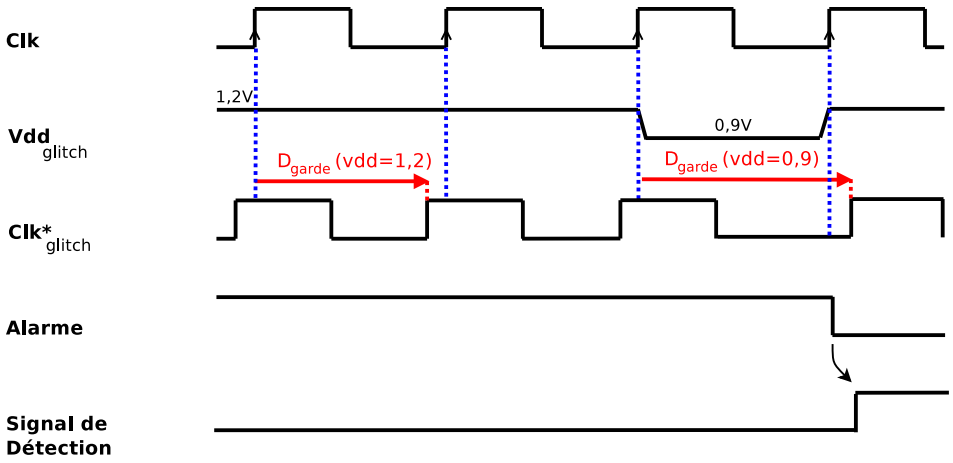

FIGURE 2.20 – Détecteur soumis à une perturbation visant à augmenter les temps de propagation

Grâce à l'utilisation de *hardmacros*, chaque instanciation du détecteur est matériellement identique. Cependant, à cause des variations dans le processus de fabrication à l'intérieur du circuit [Sedcole 2006], les différents détecteurs peuvent avoir des délais de garde légèrement différents.

Ce détecteur a été conçu pour être indépendant de l'implémentation matérielle du circuit qu'il protège. C'est à dire que son délai de garde ne dépend pas de l'état du reste du circuit. Il protège donc théoriquement l'algorithme cryptographique contre les attaques telles que la FSA ou la DBA.

2.5 Conclusion

Dans un premier temps, une implémentation matérielle de l'algorithme de chiffrement AES a été décrite. Elle servira de cible aux injections de fautes par violation de contraintes temporelles.

Ensuite, les bancs d'injections pouvant conduire à ce genre de violations ont été présentés.

Un voltmètre embarqué a aussi été décrit, celui-ci nous permettra, dans le chapitre 3, d'effectuer une analyse plus fine des perturbations internes induites par des variations transitoires de la tension d'alimentation du circuit.

Enfin, un détecteur conçu pour détecter les violations de contraintes temporelles a été présenté. Son efficacité face aux impulsions électromagnétiques sera discutée dans le chapitre 4. Il servira aussi d'exemple pour illustrer les fuites d'information entre blocs logiques indépendants dans le chapitre 5.

Chapitre 3

Résultats expérimentaux relatifs à l'injection de fautes par violation de contraintes temporelles

Sommaire

Les techniques d'injection statiques présentées dans le chapitre 1 seront effectivement mises en œuvre à l'aide des bancs d'injections pour prouver expérimentalement qu'elles induisent des violations de contraintes temporelles. Ensuite, les glitchs négatifs de tension seront injectés de façon empirique pour vérifier qu'ils induisent eux

aussi des violations de contraintes temporelles. Enfin, le voltmètre intégré sera utilisé pour observer les perturbations effectivement injectées dans le circuit soumis à ces glitchs de tension. Les résultats ainsi obtenus nous conduirons à étendre l'étude aux glitchs positifs de tension et à l'injection de deux glitchs consécutifs afin d'améliorer notre compréhension du mécanisme d'injection.

L'implémentation matérielle de l'AES servira de cible d'étude.

3.1 Résultats d'injections statiques

Les augmentations de la fréquence induisent des injections de fautes par violations de contraintes temporelles. Elles seront donc utilisées comme références et les fautes obtenues seront comparées à celles induites par les deux autres méthodes d'injection statiques : diminution de la tension et augmentation de la température. Ces trois méthodes ont un effet global sur le circuit. Comme présenté dans la section 1.3.1, une faute induite par violation des contraintes temporelles peut avoir un comportement non-déterministe. Aussi, dans le cadre de cette étude, le stress (en fréquence, en tension ou en température) est augmenté petit à petit jusqu'à l'apparition de la première faute en sortie de l'AES.

3.1.1 Augmentation statique de la fréquence

Pour chaque couple {Message, Clé secrète}, la première faute obtenue par augmentation de la fréquence correspond au chemin le plus critique de la cible pour ces valeurs de message clair et de clé secrète considérées. Les temps de propagation dans la logique combinatoire dépendent des données traitées.

Pour réaliser les expériences suivantes, 10 000 couples {Message, Clé secrète} choisis aléatoirement ont été utilisés. Pour chaque couple la procédure suivante a été appliquée (avec $Nb_{couple} = 10000$) :

Algorithme 1 Procédure d'injection de faute en statique

Pour 0 à Nb_{couple} , **faire :**
  Envoie du Message et de la clé secrète
  Premier chiffrement dans les conditions nominales de fonctionnement
  Récupération du chiffré correct ($C_{correct}$) qui servira de référence
  $Taux_{faute} = 0$
  Tant que $Taux_{faute} \neq 0$, **faire :**
    Augmentation du stress appliqué au circuit cible
    Pour 0 à 100 , **faire :**
      Chiffrement
      Récupération du chiffré (C_{temp})
      Si $C_{temp} \neq C_{correct}$ **alors**
        $Taux_{faute} = Taux_{faute} + 1$
      Fin de si
    Fin de boucle pour
  Fin de boucle tant que
Fin de boucle pour

Le pas d'incrémentation de la fréquence a été arbitrairement fixé à 200 kHz.

Si le stress est assez important, le résultat obtenu en sortie de l'AES perturbé peut être différent de celui obtenu en fonctionnement normal. Nous avons pour cela présupposé que les fautes injectées par violations de contraintes temporelles étaient injectées sur le chemin de calcul de l'AES et non pas sur le chemin de calcul des sous-clés. En effet, la structure implémentée pour les bloc de calcul de l'AES entraîne des temps critiques assez important (environ 8ns) par rapport aux temps de calcul des sous-clé (environs 4ns).

Ce résultat fauté peut être déchiffré théoriquement en utilisant la même clé secrète. Ensuite, les états intermédiaires du chiffrement correct et du chiffrement fauté sont comparés afin de retrouver où a été injectée la faute.

L'élément : {Message (M), Clé secrète (K), Chiffré correct (C), Période d'horloge (T_{stress}), Chiffré fauté (E), Ronde fautée (R), Bit fauté (B), Taux d'injection (%)} est sauvegardé. La figure 3.1 représente un élément de la bibliothèque qui pourrait être obtenu. Ici, la faute a été injectée lors de l'échantillonnage du bit 49 de la ronde 9 de l'AES.

M	32	43	$F6$	$A8$	88	$5A$	30	$8d$	31	31	...	07	34
K	$2b$	$7e$	15	16	28	ae	$d2$	$a6$	ab	$f7$	...	$4f$	$3c$
C	39	25	84	$1d$	02	dc	09	fb	dc	11	...	$0b$	32
T	$7800ps$												
E	39	25	84	$1d$	02	**0e**	09	fb	dc	11	...	$0b$	32
R	9^{ieme}												
B	49^{ieme}												
%	5%												

FIGURE 3.1 – Élément de la bibliothèque de référence "statique"

Quand deux chemins critiques sont dans leur zone de non déterminisme pour un stress identique, plusieurs fautes distinctes peuvent être injectées. Si plusieurs fautes sont injectées pour un stress identique c'est que le chemin critique relatif à ce couple {Message, Clé secrète} n'est pas beaucoup plus critique que d'autre chemins sous-critiques. Autant d'éléments sont sauvegardés qu'il y a de fautes mono-bit différentes injectées. Et un élément supplémentaire avec les données "Chiffré fauté" et "Bit fauté" non définies est sauvegardé pour compter le nombre de fautes injectées qui ne sont pas mono-bit.

M	32	43	$F6$	$A8$	88	$5A$	30	$8d$	31	31	...	07	34
K	$2b$	$7e$	15	16	28	ae	$d2$	$a6$	ab	$f7$	...	$4f$	$3c$
C	39	25	84	$1d$	02	dc	09	fb	dc	11	...	$0b$	32
T_{clk}	$7800ps$												
E	39	25	84	**ce**	02	dc	09	fb	dc	11	...	$0b$	32
R	9^{ieme}												
B	1^{er}												
%	2%												

FIGURE 3.2 – Autre élément de la bibliothèque de référence "statique" (en cas de double injection)

La figure 3.2 représente une seconde faute injectée pour le même stress et le même couple {Message, Clé secrète} que ceux considérés dans la figure 3.1 mais dans ce cas une autre faute a été injectée lors de l'échantillonnage du bit 1 de la

ronde 9 de l'AES. Une observation du non-déterminisme est faite dans la section 3.1.5.

L'ensemble des éléments obtenus expérimentalement forment une bibliothèque dite de référence "statique" qui servira de base de comparaison pour les méthodes d'injection étudiées ensuite.

La bibliothèque a été construite pour 10 000 couples {Message, Clé secrète}. Les premières fautes obtenues par augmentation progressive du stress ont été en grande majorité des fautes mono-bit, à plus de 90%. La figure 3.3 présente la répartition de ces fautes selon les rondes de l'AES qui ont été affectées en premier.

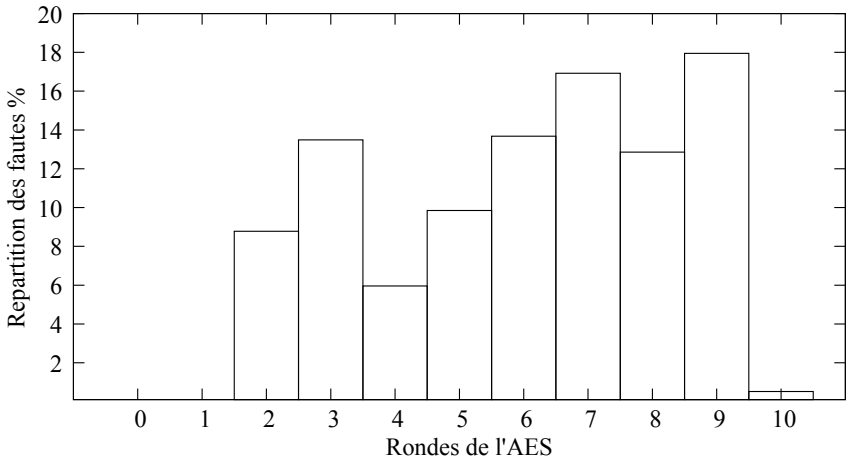

FIGURE 3.3 – Répartition des fautes mono-bit selon les rondes de l'AES

3.1.2 Diminution statique de la tension d'alimentation

Pour vérifier l'hypothèse selon laquelle les diminutions de la tension d'alimentation entraînent des violations des contraintes temporelles, des expériences ont été conduites en diminuant progressivement la tension d'alimentation de façon statique. Les résultats ainsi obtenus ont ensuite été comparés à ceux de la bibliothèque de référence "statique" construite par des augmentations progressives de la fréquence pour 10 000 couples {Message, Clé secrète}. Pour chaque couple une procédure identique à celle proposée pour la construction de la bibliothèque de référence (cf. algorithme 1) a été utilisée.

Si les fautes injectées avec ces deux moyens d'injection sont identiques, c'est que le même chemin critique a été fauté quelle que soit la méthode (1 chance sur 128 répétée 10 000 fois). Il s'agit alors d'une démonstration expérimentale de l'unicité du mécanisme d'injection.

La tension d'alimentation du circuit a donc été diminuée de façon progressive par pas de 2mV jusqu'à l'apparition d'une faute sur la sortie de l'AES. Ce protocole d'injection est illustré avec la figure 3.4. La partie inférieure de la figure représente la tension de cœur du circuit qui est diminuée de façon statique. La partie supérieure de la figure représente qualitativement les effets (en gris) de cette diminution sur les chemins critiques de chacune des rondes de l'AES pour un couple {Message, clé secrète} donnée. Dans cet exemple, une faute serait injectée en ronde 9.

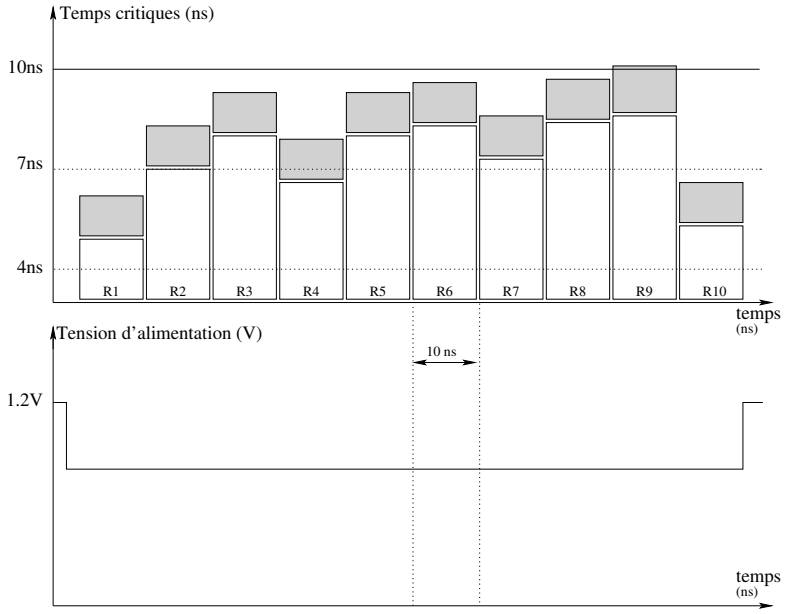

FIGURE 3.4 – Chemins critiques de l'AES soumis à une diminution statique de la tension pour un couple {Message, Clé secrète}

Pour chaque couple {Message, Clé secrète} la faute injectée (ou les fautes injectées) par augmentation de la fréquence et la faute injectée (ou les fautes injectées) par diminution de la tension d'alimentation ont été identiques : 100% des fautes obtenues avec une méthode ont été retrouvées avec l'autre méthode. Les fautes ont été induites pour des tensions statiques comprises entre 1,061V et 0,979V avec une valeur moyenne de 1,02V.

En parallèle, le banc d'injection de fautes par augmentation de la fréquence a été utilisé pour mesurer les chemins critiques relatifs à la tension appliquée au circuit. En effet, en diminuant la tension d'alimentation les temps de propagation augmentent et donc les chemins critiques relatifs à chaque couple {Message, Clé secrète} changent aussi. Diminuer (ou augmenter) la période d'horloge jusqu'à que

la faute soit injectée permet de mesurer le *slack* (la marge entre la dernière transition et la période nominale de l'horloge). La figure 3.5 représente le chemin critique en fonction de la tension d'alimentation pour 3 couples {Message, Clé secrète} différents (Couple 1, Couple 2 et Couple 3). Les résultats obtenus sur la plage de tension considérée présentent une tendance linéaire de l'évolution des temps critiques en fonction de la tension d'alimentation. Ces résultats illustrent aussi la dépendance des temps critiques aux données manipulées par l'algorithme puisque les chemins sont différents pour chacun des 3 couples {Message, Clé secrète} considérés.

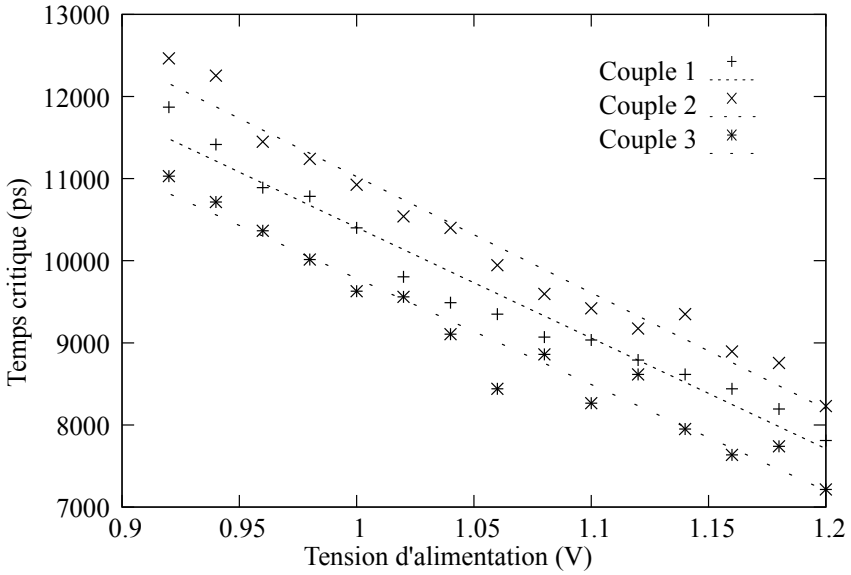

FIGURE 3.5 – Évolution des chemins critiques de l'AES en fonction de la tension d'alimentation pour 3 couples choisis {Message, Clé secrète}

3.1.3 Augmentation statique de la température

Pour vérifier l'hypothèse selon laquelle les augmentations de la température entraînent des violations des contraintes temporelles, des expériences ont été conduites en augmentant progressivement la température de façon statique suivant la même procédure que l'algorithme 1 avec $Nb_{couple} = 50$. Une bibliothèque a été construite de cette façon et comparée à la bibliothèque de référence "statique" obtenue par augmentation statique de la fréquence.

Compte tenu du caractère empirique de ce banc d'injection, les acquisitions ont été faites pour seulement 50 couples {Message, Clé secrète} différents.

Les températures considérées sont comprises entre 20°C (la température am-

biante) et 150°C. Les fautes obtenues sur cette plage de température ont été dans tous les cas identiques à celles obtenues par augmentation statique de la fréquence. Nous avons donc conclu que conformément à nos attentes, une augmentation de la température induit une augmentation des chemins critiques et finalement à des violations de contraintes temporelles sur le temps de setup (pour des températures supérieures à 110°C).

En parallèle des attaques en température, la fréquence est augmentée jusqu'à l'apparition d'une faute pour mesurer le *slack* et pouvoir tracer l'évolution des chemins critiques en fonction de la température. La figure 3.6 présente l'évolution des chemins critiques en fonction de la température pour les 3 mêmes couples {Message, Clé secrète} (Couple 1, Couple 2 et Couple 3) que dans la figure 3.5.

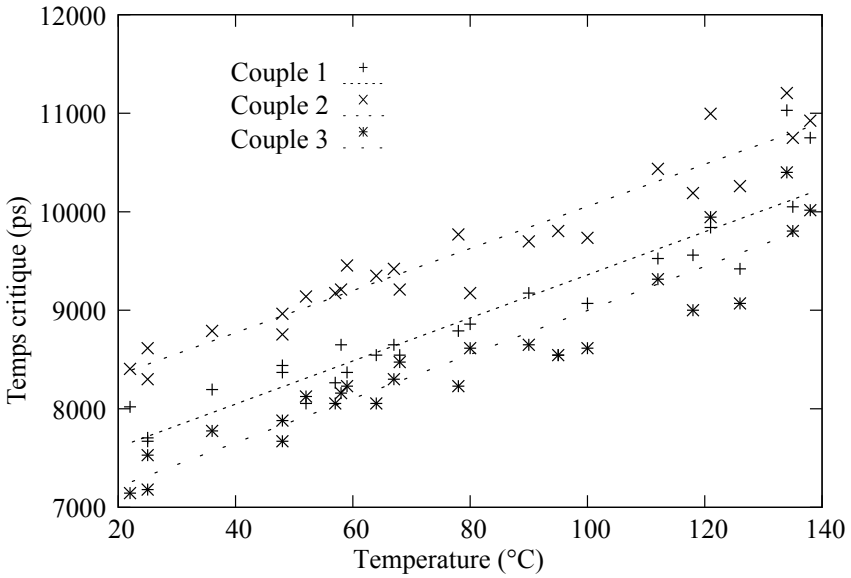

FIGURE 3.6 – Évolution des chemins critiques de l'AES en fonction la température pour 3 couples choisis {Message, Clé secrète}

3.1.4 Attaque combinée : tension et température

Dans la mesure où la diminution de la tension et l'augmentation de la température induisent les mêmes effets sur le circuit, elles peuvent être utilisées ensemble pour réduire le *slack* séparant le chemin critique de l'AES de la période d'horloge. Les acquisitions ont été faites pour les mêmes 50 couples {Message, Clé secrète} différents qui ont été utilisés pour les attaques en température seule. Le protocole suivant a été suivi :

Algorithme 2 Procédure d'injection de faute par attaque combinée en statique

Pour 0 à Nb_{couple} , **faire :**
  Envoie du Message et de la clé secrète
  Premier chiffrement dans les conditions nominales de fonctionnement (T=25°C et Vdd=1,2V)
  Récupération du chiffré correct ($C_{correct}$) qui servira de référence
  Pour Température T = 25°C à T=180°C , **faire :**
    $Taux_{faute} = 0$
    Augmentation du stress en température appliqué au circuit cible
    Tant que $Taux_{faute} \neq 0$, **faire :**
      Augmentation du stress en tension appliqué au circuit cible
      Pour 0 à 100 , **faire :**
        Chiffrement
        Récupération du chiffré (C_{temp})
        Si $C_{temp} \neq C_{correct}$ **alors**
          $Taux_{faute} = Taux_{faute} + 1$
        Fin de si
      Fin de boucle pour
    Fin de boucle tant que
  Fin de boucle pour
Fin de boucle pour

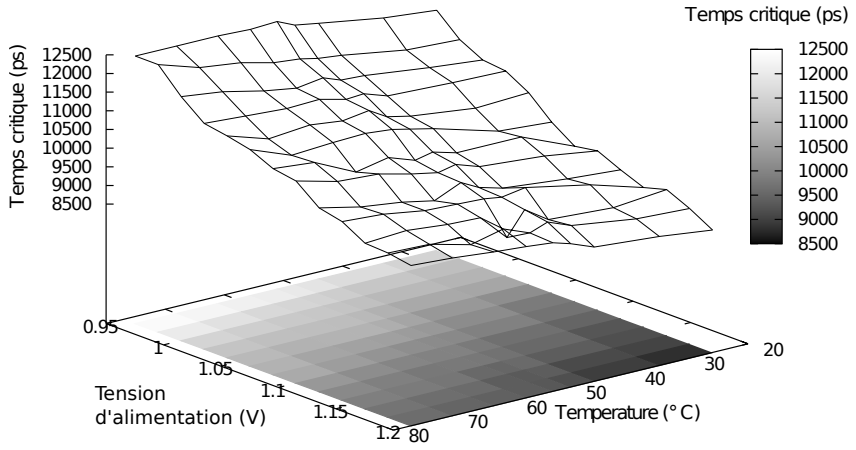

FIGURE 3.7 – Évolution d'un chemin critique de l'AES en fonction de la tension d'alimentation et de la température

Les fautes obtenues ont été identiques à celles obtenues pour chacune des méthodes considérée séparément. La figure 3.7 présente l'évolution du chemin critique relatif à un couple {Message, Clé secrète} en fonction de la température et de la tension d'alimentation.

Ce type d'attaque peut être considéré quand des détecteurs individuels sont présents sur le circuit pour surveiller indépendamment la tension et la température.

3.1.5 Observation de la zone de non-déterminisme

Comme introduit dans la section 1.3.3, l'injection de fautes en fonction du stress appliqué n'est pas toujours déterministe. Pour observer ce phénomène 3 couples {Message, Clé secrète} avec des chemins critiques assez différents ont été sélectionnés. Le premier couple est celui ayant le chemin le plus critique (parmi les 10000 couples choisis de façon aléatoire). Le second a été choisi aléatoirement. Le troisième a un chemin critique proche de la moyenne. Pour chacun de ces couples le stress a été augmenté progressivement et pour chaque stress le chiffrement a été répété 100 fois pour détecter le non-déterminisme. La figure 3.8 représente le taux d'injection de fautes sur le chemin critique de chacun de ces couples (Couple 1, Couple 2 et Couple 3) en fonction du stress statique appliqué sur la période d'horloge du circuit. Si l'on considère le couple 1 :

- Pour $T_{clk} > 8{,}8$ns, aucune faute n'est injectée.
- Pour $T_{clk} = 8{,}7$ns, le taux d'injection de fautes est de 28%.
- Pour $T_{clk} < 8{,}5$ns, le taux d'injection de fautes est de 100%.

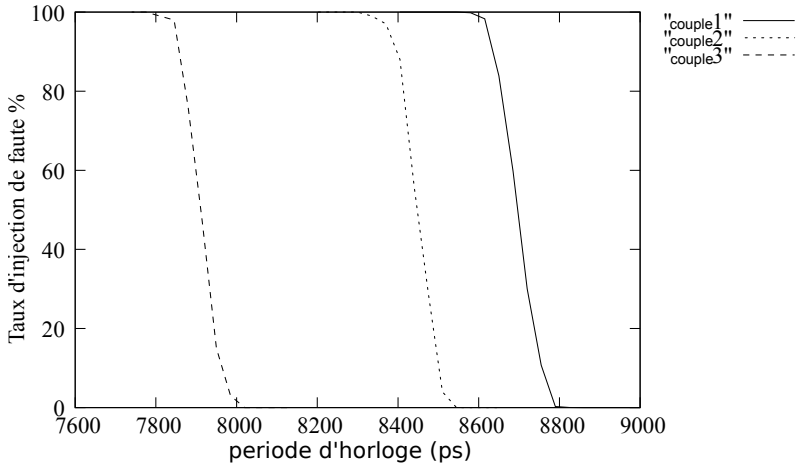

FIGURE 3.8 – Taux d'injection de fautes en fonction du stress appliqué sur 3 chemins critiques différents

Quand, pour un couple {Message, Clé secrète}, deux temps critiques sont proches, ils peuvent se trouver dans un état non déterministe pour le même stress. Alors, la première faute peut être injectée sur le chemin sous-critique avant d'être injectée sur le chemin le plus critique. En répétant le chiffrement 100 fois pour un stress donné et en mesurant les taux d'injection, l'ordre d'apparition des fautes a pu être comparé de façon plus juste.

3.1.6 Taux de fautes mono-bit

Une augmentation progressive du stress appliqué à la puce a permis d'obtenir un taux d'injection de fautes mono-bit supérieur à 90%. Sur les 10% restants , 60% des fautes ne sont pas mono-bit, les autres 40% des ces fautes sont dues à violations de contraintes de temps de setup sur deux chemins critiques appartenant à deux rondes différentes de l'AES. Ce phénomène est illustré figure 3.9 où la partie de gauche présente les temps critiques de chaque ronde de l'AES pour un couple en l'absence de stress. La partie de droite présente l'effet d'une augmentation de la fréquence : injection d'une faute mono-bit sur la ronde 7 qui entraîne un changement des temps critiques des rondes suivantes et une seconde injection de faute.

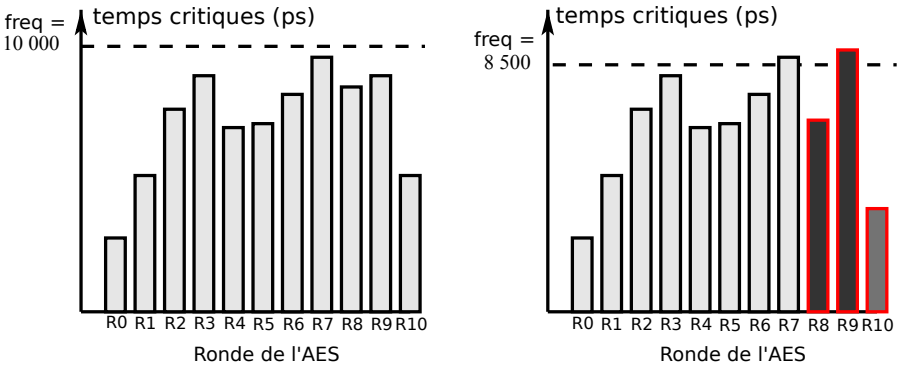

FIGURE 3.9 – Modifications du temps critique des rondes suivant l'injection d'une faute

3.1.7 Synthèse

Les expériences précédentes menées à l'aide de perturbations statiques de la fréquence, de la tension d'alimentation et de la température ont montré que ces trois méthodes partagent le même mécanisme d'injection de fautes : des violations des contraintes temporelles sur le temps de setup. De plus, ces méthodes ont un taux

d'injection de fautes mono-bit supérieur à 90% et une très grande reproductibilité.

Cependant les perturbations statiques ne permettent pas un contrôle précis de la ronde visée. Aussi pour cibler une ronde donnée, il faut changer le couple {Message, Clé secret} jusqu'à ce que le chemin critique de l'AES soit sur cette ronde et que l'injection d'une faute à cet instant n'en entraîne pas d'autres sur les suivantes pour satisfaire ainsi le modèle de faute nécessaire à la réalisation d'une DFA.

3.2 Résultats d'injections dynamiques

Afin d'améliorer la précision temporelle des injections et pouvoir viser la ronde de l'AES souhaitée, la fréquence a été augmentée dynamiquement de façon à ce que seule une période d'horloge soit diminuée. A l'aide du bancs d'injection de glitch d'horloge, la période de fonctionnement du circuit peut être diminuée par pas de 35ps sur une seule ronde.

3.2.1 Augmentation transitoire de la fréquence

Pour rappel, dans la section précédente 3.1.1 traitant de l'augmentation statique de la fréquence, seul le couple {Message, Clé secrète} était choisi. La ronde ainsi fautée était retrouvée a posteriori par simulation théorique. Cette méthode statique ne permet pas à l'attaquant de choisir une ronde en particulier. A l'inverse, dans le cas d'injections dynamiques, la ronde visée peut être choisie. De ce fait, la bibliothèque de référence "dynamique" compte environ 10 fois plus d'éléments que la bibliothèque de référence "statique" car pour chaque ronde visée, 1 ou plusieurs éléments peuvent être sauvegardés en fonction du nombre de fautes injectées.

Pour construire cette bibliothèque de référence "dynamique" un protocole similaire à celui utilisé pour construire la bibliothèque de référence "statique" (cf. algorithme 1) est appliqué. Dans ce cas, toutes les rondes de l'AES sont visées successivement sauf la ronde 0 qui est très courte et dont le temps critique est plus court que celui de la machine d'état.

Algorithme 3 Procédure d'injection de faute en dynamique

Pour 0 à Nb_{couple} , **faire :**
  Envoie du Message et de la clé secrète
  Premier chiffrement dans les conditions nominales de fonctionnement
  Récupération du chiffré correct ($C_{correct}$) qui servira de référence
  Pour Ronde visée = R1 à R10 , **faire :**
    $Taux_{faute} = 0$
    Tant que $Taux_{faute} \neq 0$, **faire :**
      Augmentation du stress appliqué au circuit cible
      Pour 0 à 100 , **faire :**
        Chiffrement
        Récupération du chiffré (C_{temp})
        Si $C_{temp} \neq C_{correct}$ **alors**
          $Taux_{faute} = Taux_{faute} + 1$
        Fin de si
      Fin de boucle pour
    Fin de boucle tant que
  Fin de boucle pour
Fin de boucle pour

Le pas de réduction de la période visée est imposé par le matériel d'injection utilisé, il est de 35ps pour une période nominale de 10ns.

Dans ces conditions, 100% des injections ont affecté la ronde visée pour un taux d'injection de fautes mono-bit toujours supérieur à 95% (les autres fautes étant des fautes multi-bits sur la même ronde dues à des temps sous critiques proches des temps critiques).

3.2.2 Diminution transitoire de tension

3.2.2.1 Injections à l'aide d'un seul générateur de glitch

Dans un premier temps, un seul générateur de glitchs (Agilent 8114A) a été utilisé afin d'injecter une diminution transitoire de la tension interne du circuit. Un glitch négatif de tension le plus court possible (10ns) a été considéré et le même protocole que celui présenté pour les augmentations transitoires de la fréquence (cf. algorithme 3) a été utilisé : l'injection du glitch est synchronisée avec la ronde visée (dans cette étude les injections ont été limités à la ronde 3, 7 et 9) et l'amplitude

du glitch injectée est augmentée progressivement jusqu'à l'apparition d'une faute.

Dans ces conditions d'injection, des fautes sont apparues pour des amplitudes négatives supérieures à 40V. Cette grandeur peut paraître extravagante comparée à la tension nominale de 1.2V mais en réalité les perturbations effectivement induites dans le circuit sont très atténuées (quelques centaines de millivolts). De plus, les fautes ainsi injectées ne perturbent pas systématiquement la ronde visée :

- Le taux d'injection de fautes mono-bit sur la ronde 3 est d'environ 7%
- Le taux d'injection de fautes mono-bit sur la ronde 7 est d'environ 68%
- Le taux d'injection de fautes mono-bit sur la ronde 9 est d'environ 88%

Les résultats obtenus pour les rondes 3 et 7 sont très loin des 90% d'injections mono-bit obtenu en statique. De plus, un grand nombre de fautes sont injectées involontairement sur les rondes suivant la ronde visée. À parti de ces résultats, nous avons donc émis l'hypothèse que les perturbations effectivement injectées devaient être filtrées par les plots d'alimentation du FPGA et/ou les capacités internes du circuit. Du fait de ce filtrage, les perturbations affecteraient potentiellement plusieurs rondes comme présenté figure 3.10 sur laquelle le glitch de tension délivré par le générateur est représenté en bleu, il dure 10ns et la perturbation interne supposée est représentée en pointillés rouges.

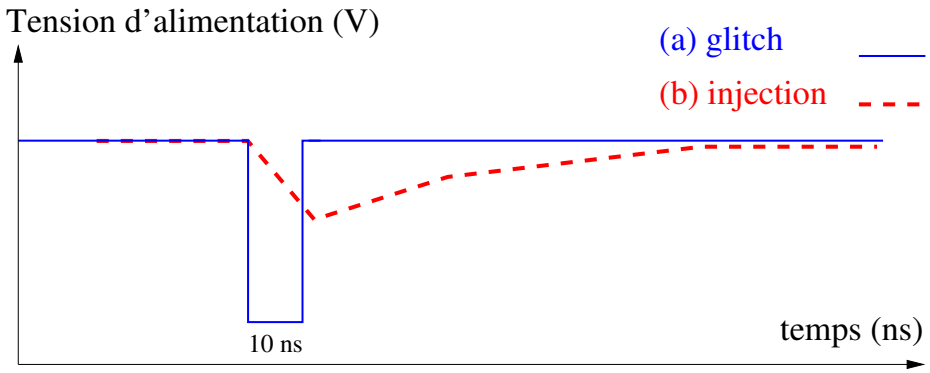

FIGURE 3.10 – Glitch envoyé et perturbation supposée être injectée dans le circuit avec un seul générateur (vue hypothétique)

Pendant l'injection, la tension sur un plot d'alimentation externe a été mesurée à l'aide l'oscilloscope (cf. figure 3.11). De nombreuses oscillations apparaissent dont une partie pourrait être induite par des rebonds dus à une mauvaise adaptation d'impédance entre le générateur et le circuit cible. L'autre partie pourrait être des oscillations effectivement présentes sur la tension de cœur du circuit.

Une représentation du même signal filtré par un filtre passe-bas théorique est aussi présentée (en pointillés sur la figure 3.11) pour donner une approximation du comportement en basse fréquence du signal à l'intérieur du circuit. Il semble effectivement que la perturbation interne dure plus longtemps qu'une seule période d'horloge. Cette observation confirme donc que plusieurs rondes de l'AES sont affectées par le glitch injecté car le signal filtré (rouge) reste en-dessous de la tension nominale (0,9V $<$ 1,2V) pendant plusieurs périodes d'horloge.

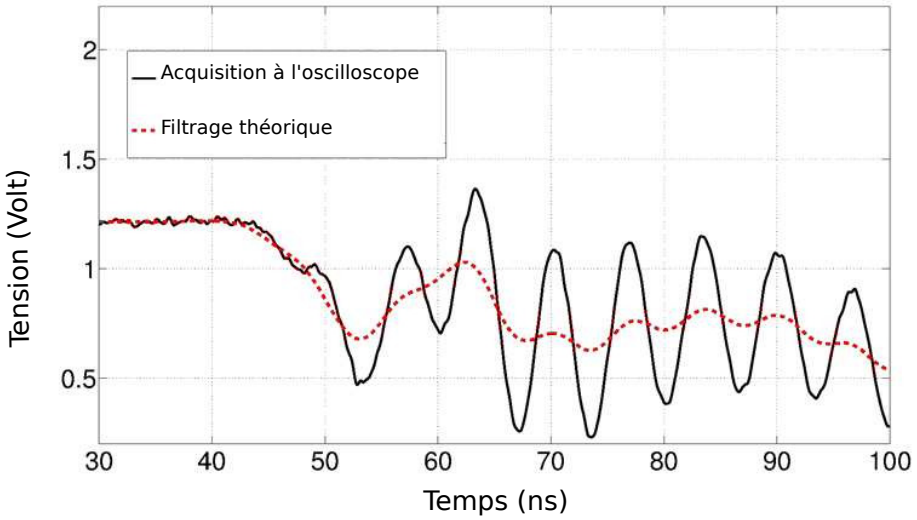

FIGURE 3.11 – Perturbations observées à l'aide d'un oscilloscope pour 1 seul glitch injecté

3.2.2.2 Injection à l'aide de deux générateurs de glitchs

Pour remédier au manque de précision temporelle constaté précédemment, un second générateur de glitch (Picosecond 10,300B) a été utilisé pour accélérer la remontée de la tension interne en délivrant un glitch positif de tension dès la fin du glitch négatif. Ainsi, les injections de fautes sur les rondes non visées sont évitées, comme illustré figure 3.12.

Une phase d'essais/erreurs a permis de déterminer de façon empirique des paramètres expérimentaux permettant d'obtenir la meilleure précision temporelle possible (taux d'injection de fautes pendant la ronde effectivement visée supérieur à 83%). Finalement, le glitch positif de "correction" retenu a une amplitude positive de 20V, dure 100ns et doit être injecté exactement quand la première injection se

termine, soit 10ns après, comme illustré figure 3.12.

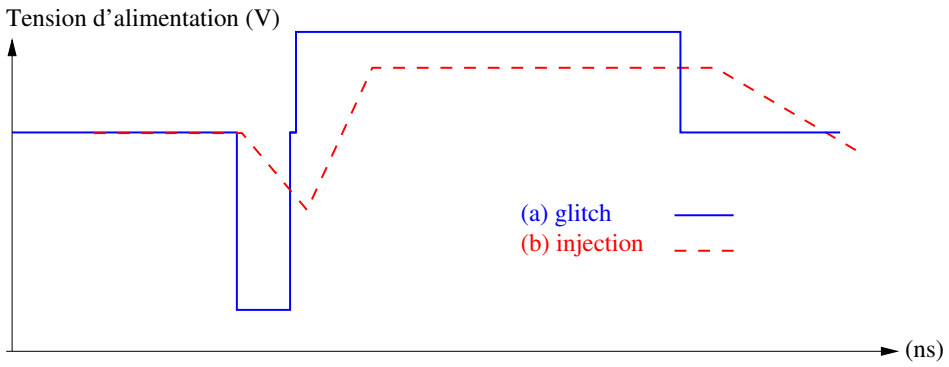

FIGURE 3.12 – Glitchs envoyés et perturbations réellement injectées dans le circuit avec deux générateurs (vue hypothétique)

La figure 3.13 présente la tension d'alimentation mesurée sur le plot d'alimentation du FPGA à l'aide d'un oscilloscope pendant l'injection des perturbations. Cette fois encore, de nombreuses oscillations peuvent être observées. Une représentation du même signal filtré permet à nouveau d'avoir une approximation du comportement en basses fréquences de la tension de cœur du circuit. Il apparaît que la perturbation négative est maintenant beaucoup plus courte (environs 10ns).

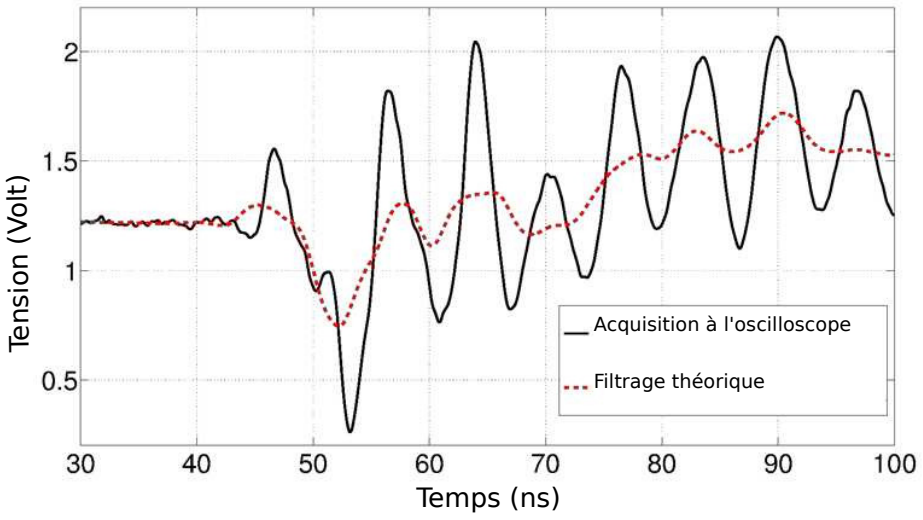

FIGURE 3.13 – Perturbations observées à l'aide d'un oscilloscope pour 2 glitchs injectés

3.2.2.3 Injection à l'aide de deux générateurs de glitchs et d'une diminution statique de la tension

Les chemins critiques de l'AES étant dépendants des données traitées, les paramètres d'injection doivent être ajustés pour chaque couple {Message, Clé secrète} et pour chaque ronde attaquée afin que le taux d'injection de fautes mono-bit soit optimal. Cette étape manuelle de paramétrage étant très longue et fastidieuse une approche plus systématique a été choisie à savoir une modification statique a été ajoutée aux perturbations dynamiques communes à toutes les injections.

Après une nouvelle phase d'essais/erreurs de nouveaux paramètres expérimentaux ont été sélectionnés empiriquement :

- Un glitch court d'injection (amplitude : -22V et durée : 10ns),
- Un glitch de "correction" (amplitude : +8V et durée : 100ns).

Ensuite la composante continue de ces glitchs a été diminuée progressivement jusqu'à l'injection d'une faute en suivant le même protocole que celui utilisé pour l'augmentation statique de la fréquence. Le pas de réduction de la composante continue de la tension d'alimentation est de 20mv.

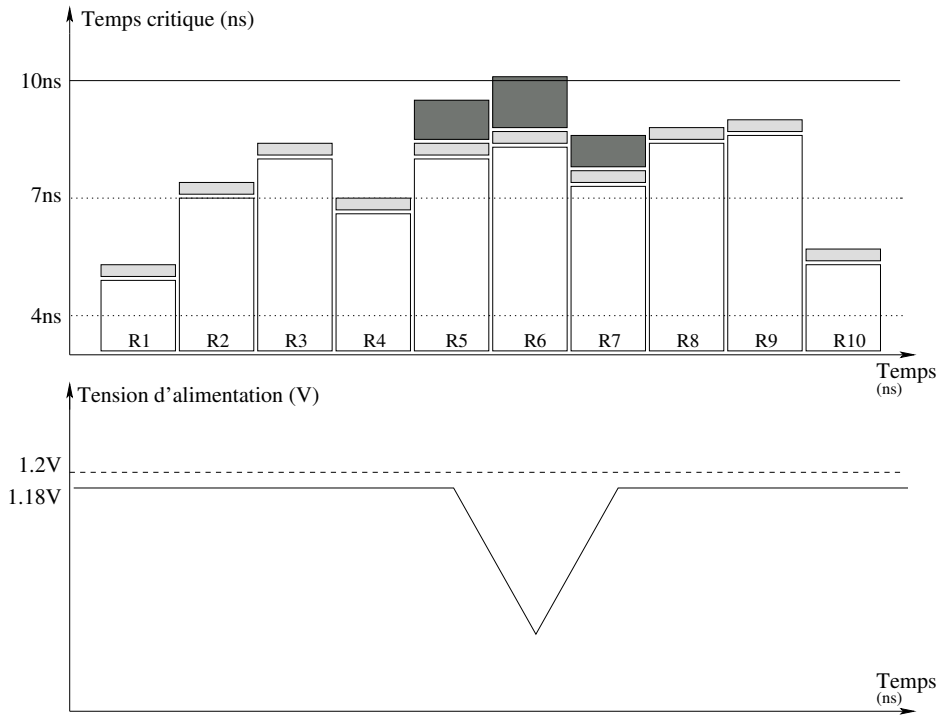

FIGURE 3.14 – Superposition des effets d'un glitch et d'une diminution statique de la tension

Cette protocole d'injection est illustrée figure 3.14 sur laquelle la partie légèrement grisée représente l'augmentation qualitative des temps critiques due à la diminution statique de la tension. La partie gris foncé représente l'augmentation qualitative des chemins critiques due aux glitchs injectés. Dans cet exemple une faute a été injectée sur la ronde 6.

Avec l'utilisation d'un double glitch le taux de fautes mono-bit injectées sur la ronde visée est supérieur à 83%.

- Les 83% de fautes injectées sur la ronde visée correspondent exactement aux fautes injectées à l'aide de glitchs d'horloge,
- Environ 6% des fautes injectées affectent des chemins sous-critiques de la ronde visée (faute mono-bit sur le second ou troisième chemin le plus critique obtenu par attaque en fréquence).
- 7% sont des injections mono-bit affectant les rondes voisines.
- Le reste des fautes ne sont pas mono-bit.

Les fautes observés sont dépendantes des données traitées par l'AES et présentent un comportement non-déterministe. Nous avons donc conclu qu'il s'agit dans les deux cas du même mécanisme d'injection lié aux violations de contraintes temporelles sur le temps de setup. D'une part, la violation de chemins sous-critiques nous laisse penser que l'injection de glitchs de tension n'affecte pas tout le circuit exactement de la même façon et que certains sous domaines d'alimentation du FPGA sont plus affectés que d'autres, ce qui pourrait expliquer les différences avec les injections par glitch d'horloge.

D'autre part, les injections sur les rondes voisines s'expliquent par la dépendance aux données des temps de propagation (cf. figure 3.15). En effet, il est possible que le chemin critique de la ronde visée soit relativement court par rapport aux chemins critiques de ses rondes voisines. Dans le même temps la perturbation effective du circuit est plus longue qu'une période d'horloge et n'affecte pas seulement la ronde visée. De ce fait, il se peut que dans certains cas, la première faute soit injectée sur la mauvaise ronde. Dans l'exemple illustré figure 3.15, l'injection est synchronisée sur la ronde 4 de l'AES mais les effets de la perturbation entraînent une augmentation des temps de calcul des rondes 3 et 5. L'augmentation du temps critique sur la ronde 5 est suffisante pour qu'une faute soit injectée sur cette ronde plutôt que sur la ronde 4.

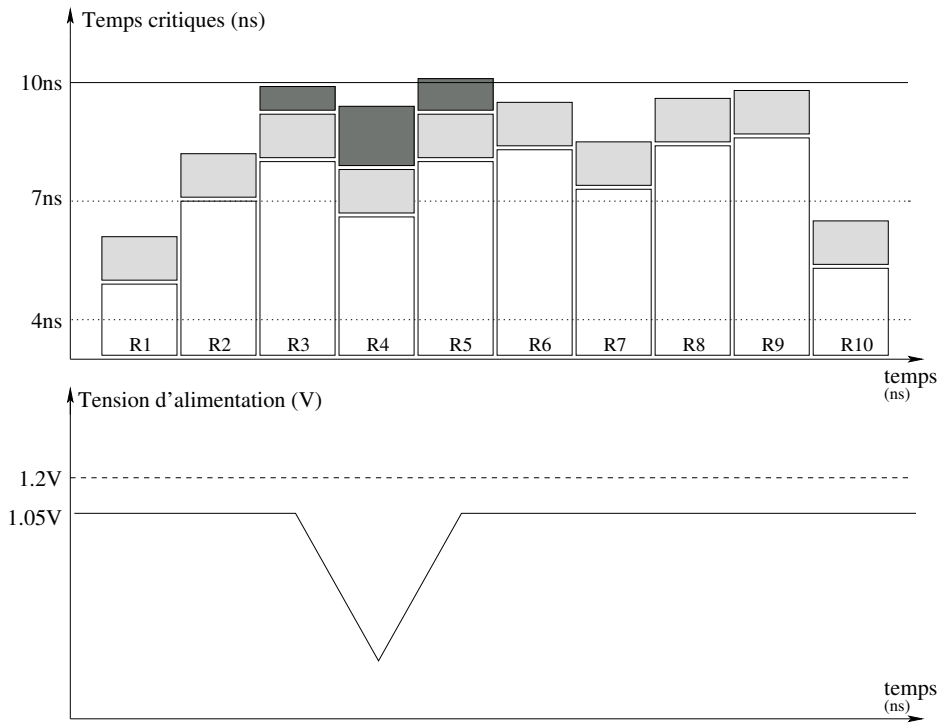

FIGURE 3.15 – Injection d'une faute dans une ronde voisine de la ronde ciblée (La ronde 5 est fautée au lieu de la ronde 4)

3.3 Observations de la tension interne à l'aide du voltmètre intégré

Dans la section précédente des paramètres empiriques d'injection de fautes à l'aide de variations transitoires de la tension d'alimentation ont été présentés. Dans un soucis de compréhension et d'amélioration nous allons, dans cette sous section, observer grâce à un voltmètre intégré la tension du cœur du FPGA soumis à différents glitchs de tension.

3.3.1 Analyse de l'effet d'un glitch négatif de tension

Les effets d'un glitch négatif sur la tension interne du FPGA ont été observés à l'aide du voltmètre intégré présenté dans la section 2.3. Les paramètres du glitch négatif de tension considéré son les suivants :

- Durée d'injection : 400ns
- Amplitude : -14V

– Composante continue : 1,7V

La composante continue de ce glitch a été relevée de 1,2V à 1,7V afin que l'ensemble des perturbations induites sur la tension interne du circuit cible soit dans la plage de fonctionnement optimale du voltmètre intégré.

Les résultats de cette observation sont illustrés figure 3.16 : la tension interne mesurée par le voltmètre présente deux trains d'oscillations amorties dont la pseudo période avoisine les 100ns. Ces oscillations amorties sont synchronisées avec les fronts descendant et montant du glitch de tension injecté. L'intervalle de temps entre les deux oscillations principales est de 400ns, ce qui correspond au temps d'injection du glitch considéré. La première oscillation négative observée, induite par le front descendant du glitch négatif de tension, a une amplitude de 400mV et une durée de 50ns environ.

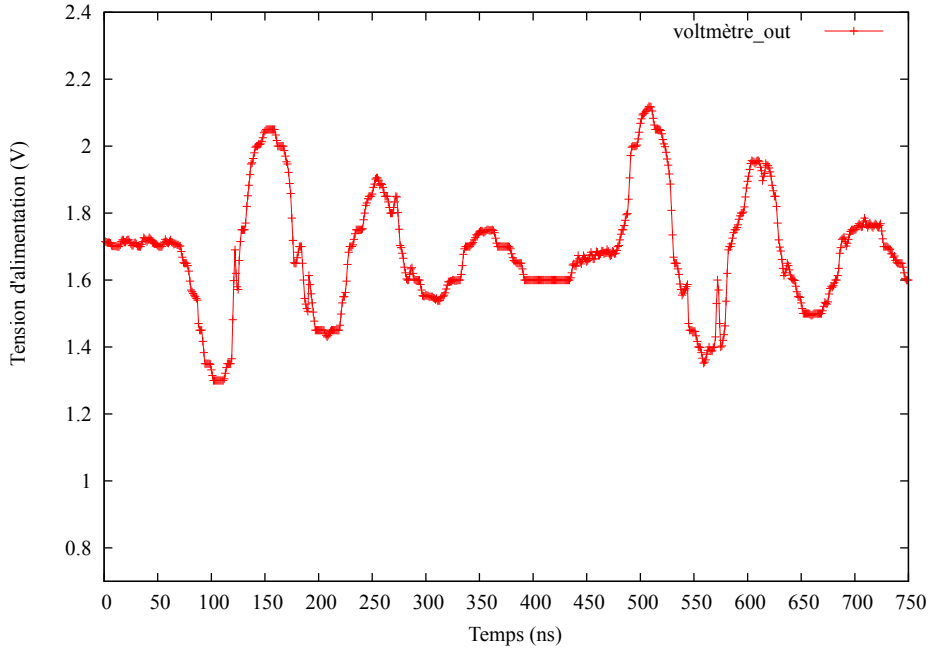

FIGURE 3.16 – Perturbation de la tension interne du FPGA pour un glitch négatif : (400ns , -14V)

Des profils de tensions identiques ont été mesurés pour d'autres paramètres d'injection. L'analyse de ces observations nous a permis de conclure qu'un front descendant injecté par une générateur de tension induit des oscillations amorties commençant par une oscillation négative sur la tension interne du circuit et qu'un front montant injecté par une générateur de tension induit des oscillations amorties

commençant par une oscillation positive sur la tension interne du circuit. Les perturbations internes sont observés à partir de 80ns après le signal de synchronisation entre le voltmètre et le générateur de glitch : c'est le temps de réponse du générateur.

De nouvelles expériences ont été faites en utilisant exactement les mêmes paramètres d'injection mais en visant cette fois l'implémentation de l'AES comme illustré dans la figure 3.17. Le signal de synchronisation envoyé par le FPGA (330ns avant le début du calcul) permet de synchroniser les calculs avec l'instant d'injection des perturbations sur la tension interne du FPGA. De cette façon la première oscillation négative induite par le front descendant du glitch sur la tension interne du FPGA a pu être synchronisée avec les différentes rondes de l'AES. L'instant d'injection a été incrémenté par pas de 5ns. Pour chaque instant d'injection la composante continue du glitch a été progressivement diminuée à partir de 1.7V et jusqu'à ce qu'une faute apparaisse.

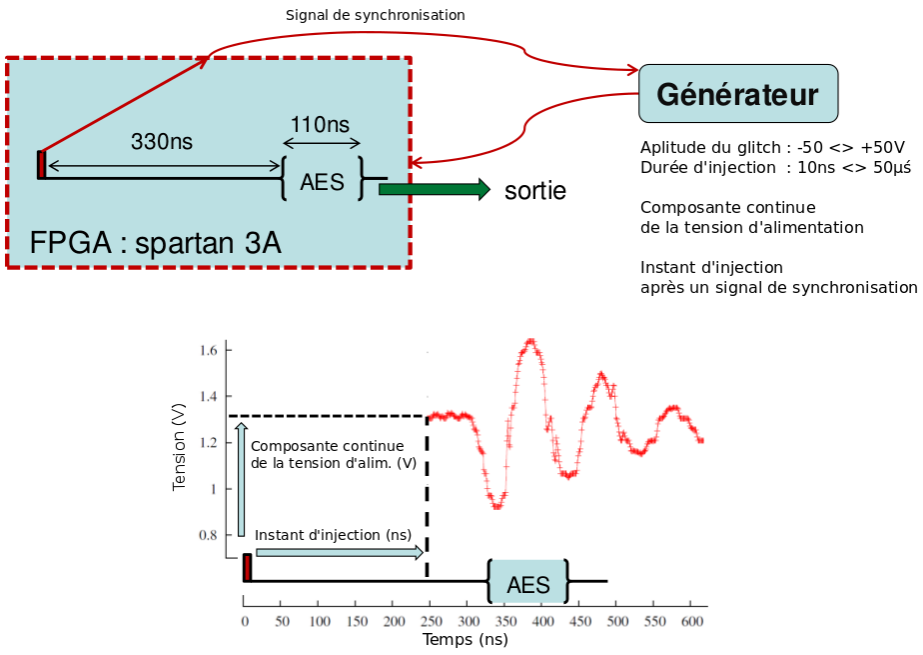

FIGURE 3.17 – Synchronisation de la première oscillation négative avec l'AES

- L'injecteur de glitch a un temps de réponse de 80ns après la réception du signal de synchronisation envoyé par le FPGA,
- La perturbation négative maximale injectée sur la tension interne est 25ns après le front descendant du glitch envoyé par le générateur (un quart de la

pseudo période des oscillations amorties),

– Le signal de synchronisation du FPGA est envoyé 330ns avant la ronde 0 de l'AES.

La figure 3.18 présente la composante continue à partir de laquelle une faute a été injectée en fonction de l'instant d'injection pour un couple {Message, Clé secrète} choisi aléatoirement. La couleur indique sur quelle ronde de l'AES la faute a été injectée. Ici, seule la précision temporelle des injections est observée, la valeur de la faute injectée n'est donc pas indiquée.

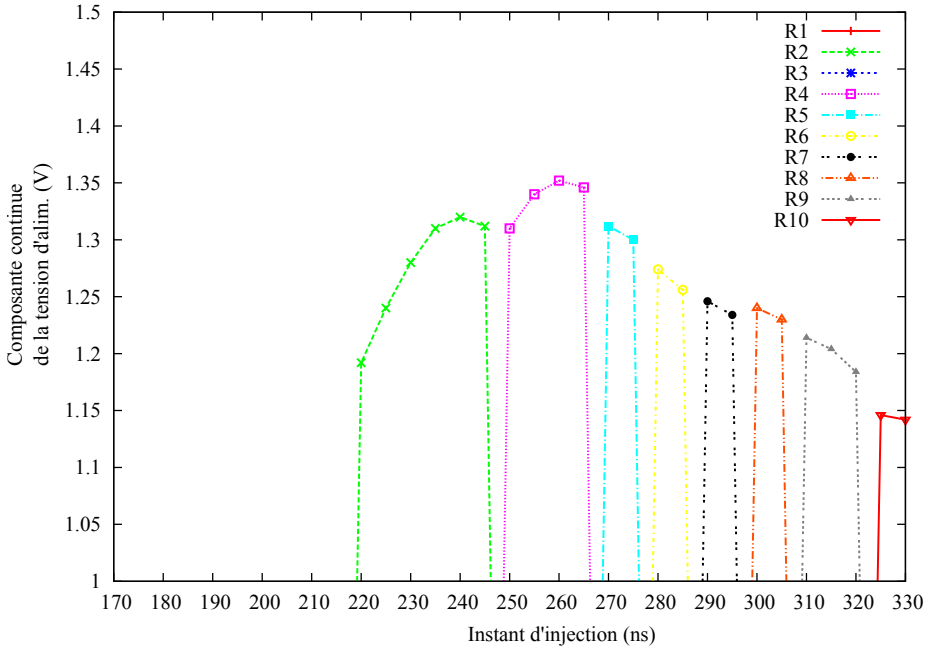

FIGURE 3.18 – Précision temporelle d'injection avec un glitch négatif : (400ns , -14V)

En tenant compte des temps de synchronisation entre le FPGA et le générateur de glitch il apparaît que les injections sont synchronisées avec la perturbation négative maximale injectée sur la tension interne. De plus, les fautes injectées sur les rondes visées étaient mono-bit et identiques à celles injectées par diminution de la période d'horloge. Aussi nous avons confirmé que les fautes étaient induites par des violations de contraintes temporelles sur le temps de setup dues aux perturbations négatives de la tension interne du circuit.

Pour le couple {Message, Clé secrète} considéré dans la figure 3.18, aucune faute n'a été injectée sur la ronde 3 en utilisant un glitch (400ns , -14V). Ceci s'explique

par le fait que les chemins critiques des rondes 2 et 4 peuvent être plus sensibles à l'injection de fautes (temps critiques plus longs) que le chemin critique de la ronde 3 et que la perturbation interne affecte plusieurs rondes de l'AES à la fois pouvant induire une faute sur les rondes voisines comme illustré figure 3.15.

3.3.2 Analyse de l'effet d'un glitch positif de tension

Les glitchs positifs de tension permettent aussi en pratique d'injecter des fautes mais, à notre connaissance, aucune explication rigoureuse n'a été proposée dans la littérature concernant les mécanismes associés. Une augmentation de la tension statique ne devrait pas permettre l'injection de fautes via la violation de contraintes temporelles sur le temps de setup puisqu'une augmentation statique de la tension réduit les temps de propagation des données. Cela dit, au vu des résultats précédents, il semble possible que des glitchs positifs entraînent aussi des oscillations négatives sur la tension de cœur du FPGA et puissent donc induire des violations de contraintes sur le temps de setup.

Les effets d'un glitch positif sur la tension interne du FPGA ont donc été observés à l'aide du voltmètre intégré. Les paramètres du glitch positif de tension considéré sont les suivants :

- Durée d'injection : 400ns
- Amplitude : +14V
- Composante continue : 1,7V

La figure 3.19 présente la tension interne du FPGA mesurée à l'aide du voltmètre intégré quand un glitch positif est injecté. Les perturbations induites par ce glitch sont très similaires à celles induites par le glitch négatif présentées figure 3.16. Dans le cas d'un glitch positif, les oscillations amorties commencent par une oscillation positive correspondant au front montant (contrairement à un glitch négatif qui entraîne une première oscillation négative correspondant à son front descendant).

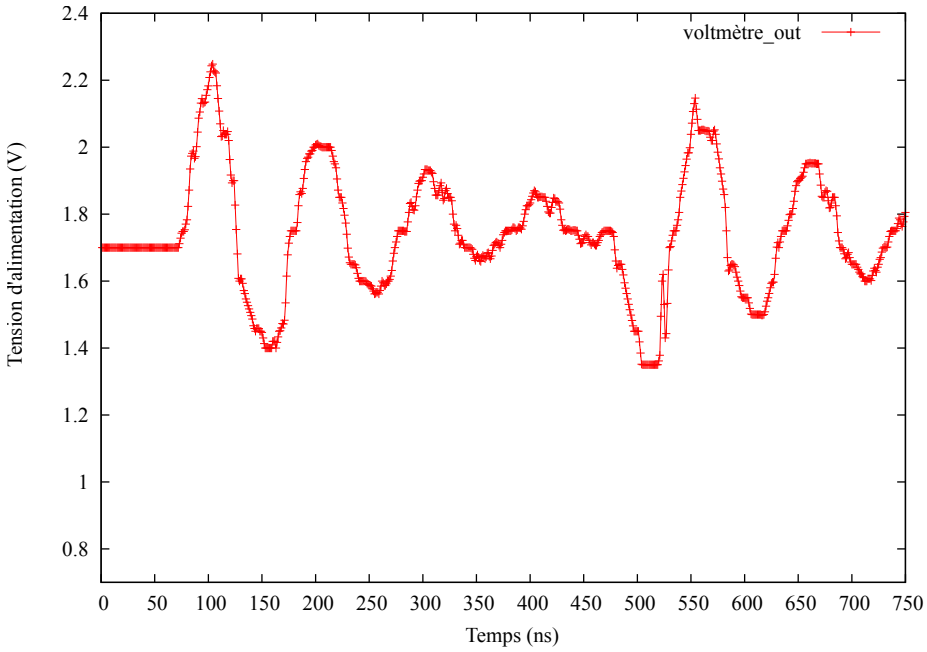

FIGURE 3.19 – Perturbation de la tension interne du FPGA pour un glitch positif : (400ns , +14V)

Le résultat principal de cette expérience est donc que les glitchs positifs induisent aussi des oscillations amorties sur la tension interne du circuit. Les oscillations négatives peuvent donc potentiellement être utilisées pour l'injection de fautes par violation de contraintes temporelles sur le temps de setup. Pour vérifier ces hypothèses, de nouvelles expériences ont été faites en utilisant exactement les mêmes paramètres d'injection (400ns , +14V) mais en visant l'implémentation de l'AES.

Comme précédemment, l'instant d'injection a été incrémenté par pas de 5ns et pour chacun de ces instants d'injection la composante continue du glitch a été réduite progressivement jusqu'à l'apparition d'une faute.

La figure 3.20 présente la composante continue à partir de laquelle une faute a été injectée en fonction de l'instant d'injection pour le même couple {Message, Clé secrète} que celui étudié dans la section 3.3.1. La couleur indique sur quelle ronde de l'AES la faute a été injectée. Cette fois encore les fautes injectées ont été identiques à celles injectées par glitch d'horloge.

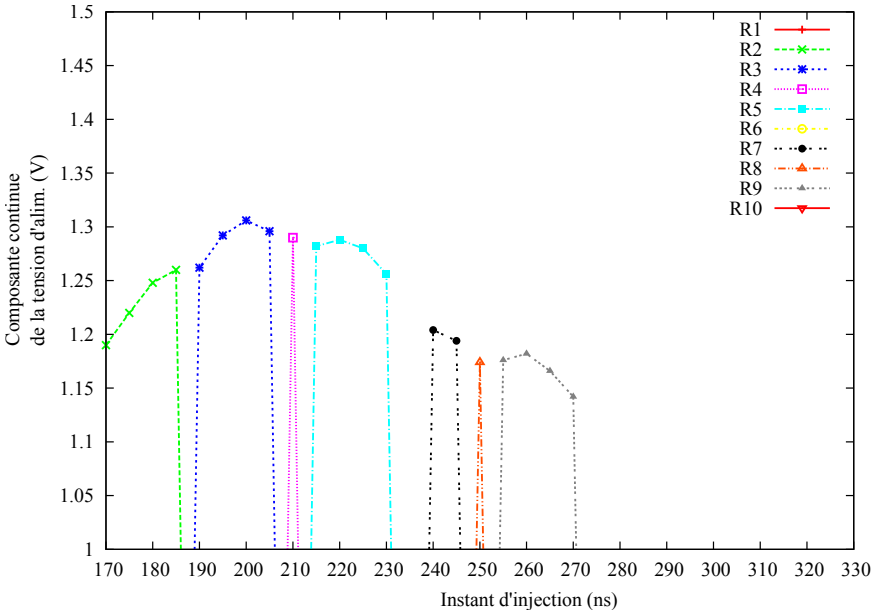

FIGURE 3.20 – Précision temporelle d'injection avec un glitch positif : (400ns , +14V)

En tenant compte des temps de synchronisation entre le FPGA et le générateur de glitch il apparaît que les injections sont synchronisées avec la première oscillation négative injectée sur la tension interne. Cette fois, la première oscillation négative est 50ns plus tard que la première oscillation négative induite par le glitch négatif étudié dans la section 3.3.1. De plus, les fautes injectées sur les rondes visées sont mono-bit et identiques à celles injectées par diminution de la période d'horloge.

Pour le couple {Message, Clé secrète} considéré aucune faute n'a été injectée sur la ronde 6 et les rondes 4 et 8 ont été fautées sur des plages de délai réduites. Les différences entre les résultats obtenus avec le glitch négatif et le glitch positif s'expliquent par la différence de forme des perturbations internes induites. Dans le cas du glitch positif (400ns , +14V) l'oscillation négative est légèrement plus large (45ns) que la première oscillation négative induite par le glitch négatif (400ns , -14V).

Ces résultats nous ont permis de confirmer que les fautes injectées à l'aide d'un glitch positif sont induites par des violations de contraintes temporelles sur le temps de setup dues aux perturbations négatives sur la tension interne du circuit.

3.3.3 Amélioration de la précision temporelle

Le paramètre principal pour faire varier la forme de la perturbation est l'amplitude du glitch et sa composante continue. Les temps de montée et descente des glitchs injectés pourraient probablement être des paramètres influents sur la formes des perturbations induites sur la tension interne du circuit mais ils ne sont pas réglables sur les générateurs utilisés.

Cependant en choisissant bien la durée d'injection d'un glitch il est possible de faire se superposer les trains d'oscillations dus respectivement à son front montant et à son front descendant. Cette superposition des deux trains d'oscillations amorties peut permettre de compenser certaines oscillations négatives par des oscillations positives (ou inversement) ou encore de faire s'additionner les effets de deux oscillations négatives (ou positives). Trois types de superpositions que nous avons étudiés (compensation, addition, et raccourcissement) sont décrits dans les sous-parties suivantes.

3.3.3.1 Compensation

Ce phénomène de compensation a pour objectif de amoindrir l'amplitude de la seconde oscillation négative du premier train d'oscillations de façon à ne conserver qu'une seule oscillation négative significative et d'éviter les injections de fautes non désirées à d'autres instants du calcul ou de la communication.

Compte tenu de la pseudo période des oscillations amorties qui est égale à 100ns, le choix d'une durée d'impulsion de 100ns permet de superposer la première oscillation positive du second train d'oscillation (dû au front montant du glitch) avec la seconde oscillation négative du premier train d'oscillation.

La figure 3.21 représente les perturbations injectées sur la tension interne du circuit soumis à un glitch négatif : (100ns, -14V). Ces perturbations ont été observées à l'aide du voltmètre intégré.

Pour un glitch d'amplitude -14V et de durée d'impulsion de 400ns, la première oscillation négative induite par le front descendant est d'environ 400mv et la seconde est d'environ 300mV (voir la figure 3.16). Avec une durée d'impulsion de 100ns, le phénomène de compensation permet de transformer cette seconde oscillation négative de 300mV en oscillation positive de 150mV.

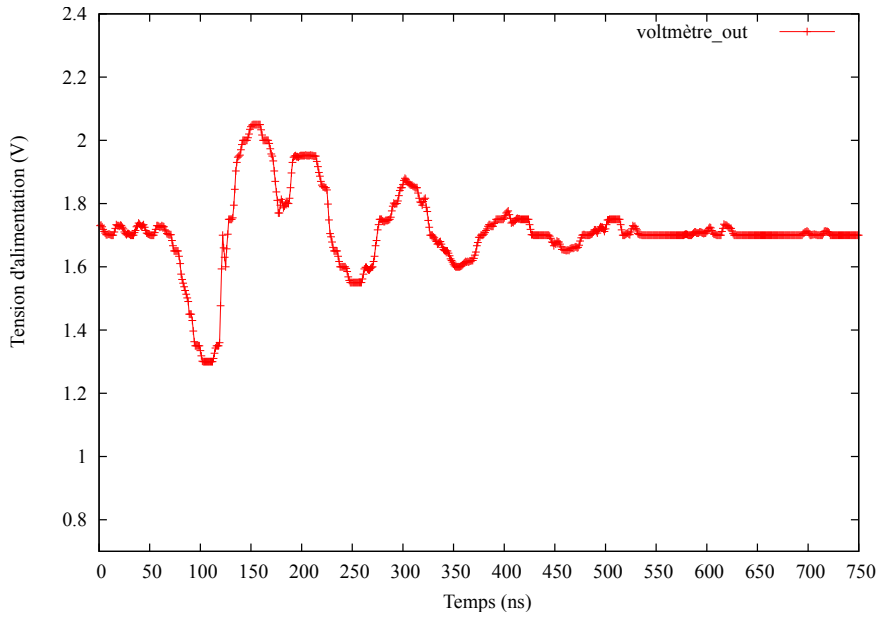

FIGURE 3.21 – Perturbation de la tension interne du FPGA pour un glitch négatif injecté : (100ns , -14V)

3.3.3.2 Addition

Ce phénomène d'addition a pour but d'injecter une perturbation assez forte avec une amplitude de glitch la plus faible possible en synchronisant deux oscillations négatives. L'objectif est donc de synchroniser la première oscillation négative du premier train d'oscillations amorties et du second train.

Compte tenu des observations précédentes et de la pseudo période des oscillations, le choix d'une impulsion positive d'une durée de 50ns permet de superposer les deux premières oscillations négatives de chaque train d'oscillations.

La figure 3.22 représente les perturbations injectées sur la tension interne du circuit soumis à un glitch positif : (50ns, +8V). Ces perturbations ont été observées à l'aide du voltmètre intégré.

Dans l'exemple présenté (figure 3.22), l'oscillation négative la plus significative est de 400mV. Cette amplitude obtenue avec un glitch d'amplitude +8V est la même que celle obtenue avec un glitch négatif d'amplitude -14V.

3.3.3.3 Raccourcissement

Le raccourcissement a pour objectif d'injecter des perturbations plus courtes et ainsi d'améliorer la précision temporelle des injections. L'objectif est donc de raccourcir la première oscillation négative d'un train d'oscillation avec la première oscillation positive du second train d'oscillations.

Le choix d'une impulsion négative d'une durée de 10ns permet de superposer la première oscillations négatives induite par le front descendant du glitch avec la première oscillation positive induite par le front montant du glitch.

Cela dit, l'oscillation significative est raccourcie avant d'avoir atteint sa valeur maximale. Du fait de ce raccourcissement, il faut que le glitch injecté ait une amplitude plus importante pour que l'amplitude de la perturbation sur la tension interne soit identique à celles obtenues avec les injections précédentes.

La figure 3.23 représente les perturbations injectées sur la tension interne du circuit soumis à un glitch négatif : (10ns, -22V). Ces perturbations ont été observées à l'aide du voltmètre intégré.

Dans l'exemple présenté (figure 3.23), la première oscillation négative d'amplitude 350mV est raccourcie (environs 15ns). Cependant un glitch d'amplitude négative -22V est nécessaire pour obtenir une première oscillation d'amplitude comparable celle obtenue avec un glitch d'amplitude négative de -14V . L'apparition d'une seconde oscillation négative d'amplitude comparable 100ns après la première est à noter car elle peut être la source d'injections de fautes non désirées.

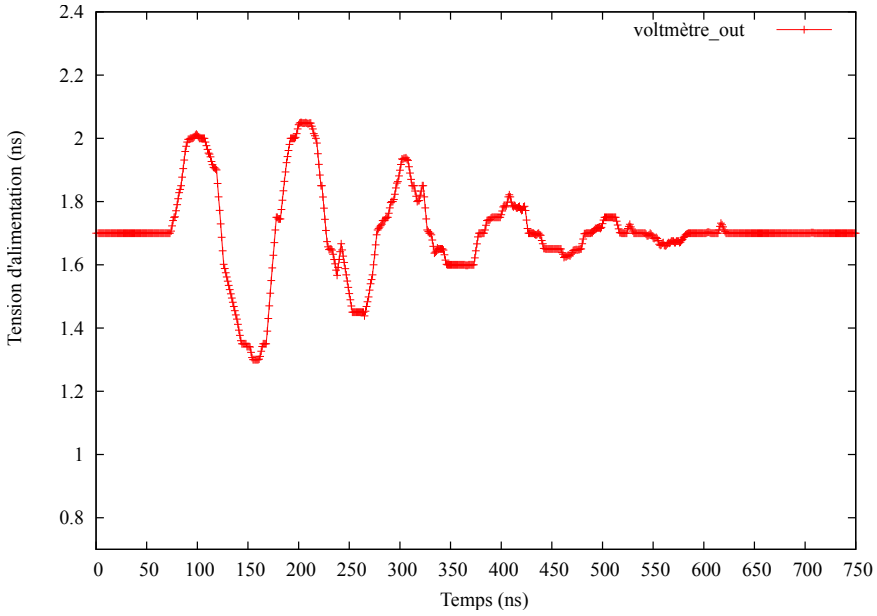

FIGURE 3.22 – Perturbation de la tension interne du FPGA pour un glitch positif injecté : (50ns , +8V)

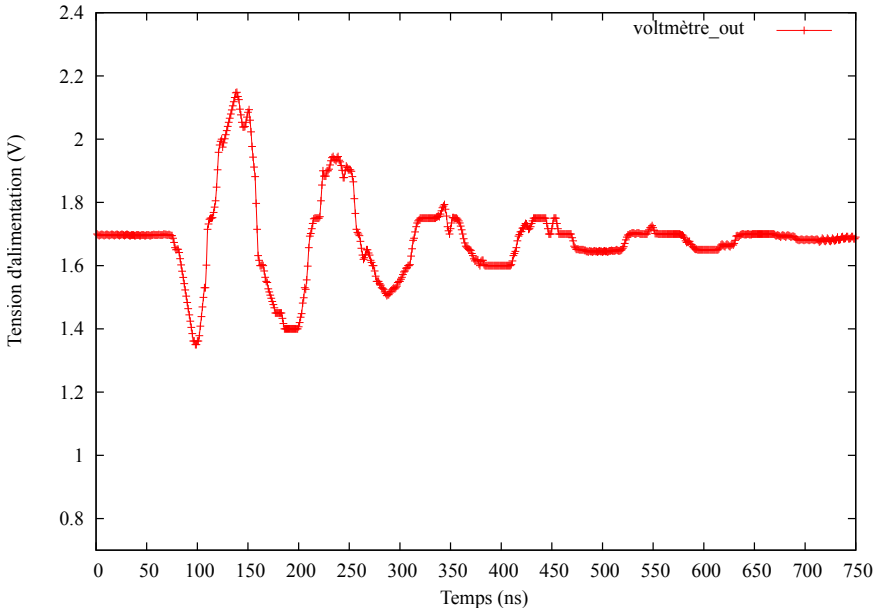

FIGURE 3.23 – Perturbation de la tension interne du FPGA pour un glitch négatif injecté : (10ns, -22V)

3.3.3.4 Résolutions temporelles des injections de fautes

Dans la partie précédente il a été observé que la durée d'injection du glitch pouvait être utilisée pour superposer les trains d'oscillations induits sur la tension d'alimentation par les fronts montant et descendant du glitch. Pour caractériser leur résolution temporelle, des fautes sur l'AES ont été injectées et comparées suivant le même protocole que celui introduit précédemment. Les figures 3.24, 3.25 et 3.26 représentent les composantes continues minimales avant qu'une faute soit injectée pour différent instants d'injection. Ces dernières ont été obtenues en considérant le même couple {Message, Clé secrète} que celui de la section 3.3.1 et pour les 3 glitchs présentés précédemment :

- Compensation : figure 3.24,
- Addition : figure 3.25,
- Raccourcissement : figure 3.26.

Les différentes couleurs représentent les rondes sur lesquelles une faute a été injectée et donc la précision temporelle obtenues avec le glitch considéré.

Les figures 3.24 et 3.25 illustrent les précisions temporelles obtenues lors de l'injection de fautes avec des glitchs ayant respectivement pour paramètres : (100ns , -14V) et (50ns , +8V). Ces deux glitchs bien qu'étant différents conduisent à des résultats en termes de précision temporelle identiques :

- Aucune faute n'a été injectée sur la ronde 6,
- Les rondes 4 et 8 ont été fautées pour un nombre réduit d'instants d'injection.

Ces résultats sont aussi très similaires à ceux obtenus pour un glitch positif ayant pour paramètres (400ns , +14V) présenté figure 3.20.

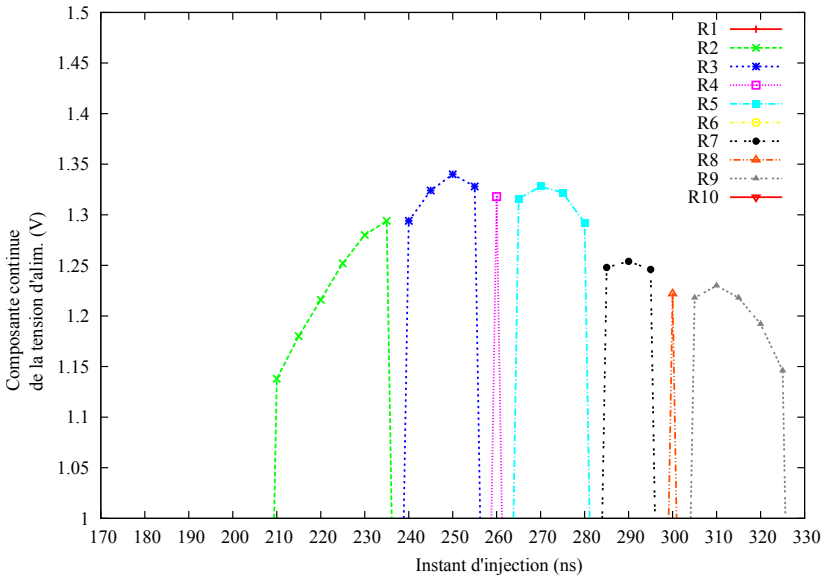

FIGURE 3.24 – Précision temporelle d'injection par compensation avec un glitch négatif : (100ns , -14V)

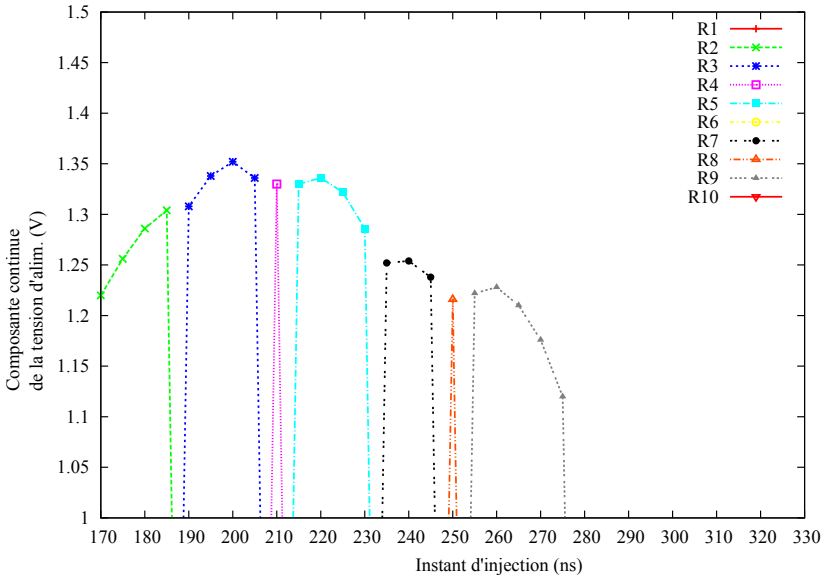

FIGURE 3.25 – Précision temporelle d'injection par addition avec un glitch positif : (50ns , +8V)

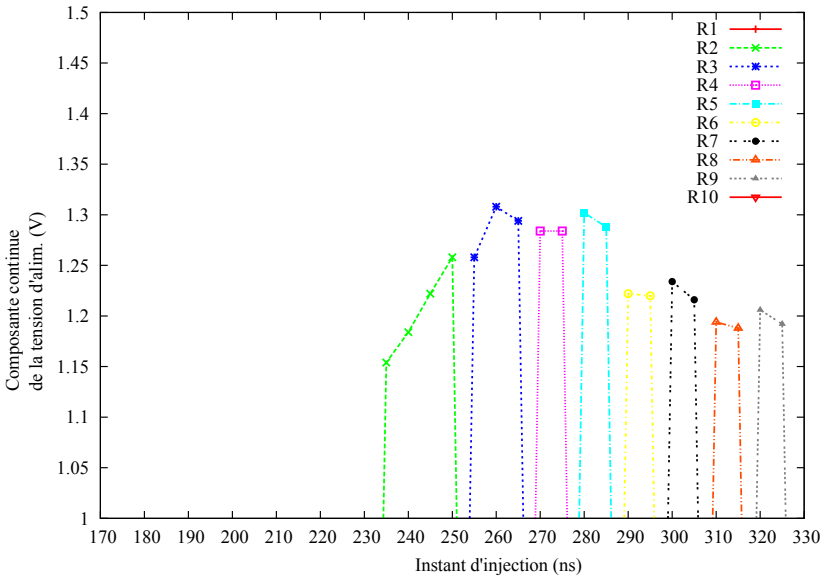

FIGURE 3.26 – Précision temporelle d'injection par raccourcissement avec un glitch négatif : (10ns , -22V)

La figure 3.26 illustre la précision temporelle obtenue lors de l'injection de fautes avec un glitch négatif ayant pour paramètres : (10ns , -22V). Conformément à nos hypothèses, l'oscillation raccourcie permet d'atteindre la meilleure résolution temporelle en effet, des fautes mono-bit ont été injectées dans toutes les rondes de l'AES. Par contre la seconde oscillation négative induite par ce glitch court est assez importante et pourrait dans certains cas avoir des effets non négligeables et indésirables sur le fonctionnement du circuit (erreur sur la communication, mauvaises interprétations des résultats).

3.3.4 Analyse a posteriori des paramètres d'injections empiriques

En analysant a posteriori les perturbations induites par le double glitch illustré de façon hypothétique par la figure 3.12, nous avons en avons déduit le gabarit suivant :

- Au temps t, glitch raccourci : (10ns, -22V),
- Au temps $t + 10ns$, glitch compensé : (100ns, +8V).

Dans ces conditions, en nous basant sur les résultats précédemment obtenus dans la section 3.3.3, les perturbations induites sur la tension interne du circuit par la superposition de ces deux injections peuvent être extrapolées.

Il doit s'agir d'une première oscillation négative raccourcie suivie d'une oscillation positive,induites par le premier glitch négatif court. Ensuite, le second glitch entraîne un seul train d'oscillations qui compense les autres oscillations induites par le premier glitch. Il ne reste donc, au final, qu'une seule oscillation négative courte.

Nous pouvons donc en conclure, a posteriori, que les paramètres d'injection trouvés empiriquement étaient bien adaptés à l'injection de fautes.

3.4 Conclusion

Dans ce chapitre les injections de fautes à l'aide de perturbations statiques ont été tout d'abord étudiées. Il a été démontré expérimentalement qu'une diminution statique de la tension d'alimentation ou qu'une augmentation statique de la température induisent les mêmes fautes dans un circuit synchrone qu'une augmentation statique de la fréquence de fonctionnement. Aussi, nous avons conclu que les mécanismes liés à ces différents moyens d'injection de fautes sont identique, il s'agit de violations des contraintes de temps de setup.

De plus, nous avons aussi observé l'évolution des temps de propagation en fonction de la tension et de la température et déduit que ces deux paramètres ont, au moins sur les plages de stress étudiées, une influence quasi-linéaire sur l'évolution des temps de propagation.

Cependant, les attaques statiques ne permettent pas de viser systématiquement une ronde de calcul choisie. Une bonne maîtrise de la précision temporelle est indispensable pour la réalisation de certaines attaques. Aussi, dans un second temps, les injections de fautes à l'aide de perturbations transitoires de tension et de fréquence ont été étudiées.

Une phase d'essais/erreurs nous a permis de converger vers une double injection (avec deux générateurs de glitchs de tension). Une autre phase de calibration nous a permis de trouver des paramètres permettant une injection de fautes avec un bonne précision temporelle. Les fautes induites par les diminutions dynamiques de tension ont été une nouvelle fois identiques à celles induites par les variations de fréquence. Nous avons donc pu en conclure qu'il s'agissait encore de violations des contraintes de temps de setup.

Enfin, l'implémentation d'un voltmètre intégré sur le FPGA cible nous a permis d'observer les perturbations réellement injectées sur la tension interne du circuit et de mieux comprendre le mécanisme d'injection de fautes. Nous avons observé

notamment qu'un glitch de tension entraîne des oscillations amorties sur la tension d'alimentation interne du circuit. Ces oscillations sont dues aux fronts montant et descendant du glitch injecté. De ce fait, les glitchs positifs comme les glitchs négatifs induisent des perturbations négatives (et positives) sur la tension interne du circuit attaqué. Les deux types de glitchs sont donc susceptibles d'induire des violations de contraintes temporelles sur le temps de setup.

L'injection de fautes par violation de contraintes temporelles à l'aide d'un glitch positif a part ailleurs été illustré expérimentalement avec un glitch positif (+8V ; 50ns) induisant des fautes identiques et une précision temporelle similaire à un glitch négatif (-14 ; 100ns).

Suite à ces observations, nous avons aussi conclu a posteriori que les paramètres trouvés empiriquement pour l'utilisation d'une double injection de glitchs étaient adaptés et fournissaient une bonne résolution temporelle.

Tous les moyens d'injection connus (à l'exception du laser) peuvent induire des violations de contraintes temporelles. Dans [Endo 2012] et [Selmane 2011] les auteurs proposent un détecteur générique basé sur un délai de garde et protégeant le circuit contre les injections de fautes par violations de contraintes temporelles. Cependant, dans le même temps, il a été évoqué dans [Poucheret 2011] et [Dehbaoui 2012b] que les impulsions électromagnétiques pouvaient induire des perturbations locales sur la tension d'alimentation interne du circuit. Aussi dans le chapitre suivant, l'efficacité d'un tel détecteur face à des injections électromagnétiques sera étudiée.

Chapitre 4

Étude des vulnérabilités du détecteur de violations de contraintes temporelles (DVCT) vis-à-vis des attaques électromagnétiques

Sommaire

Le détecteur DVCT a été proposé dans la littérature pour détecter des tentatives d'injection de fautes par violation de contraintes temporelles. Son efficacité a été montré démontré dans [Gomina 2014] vis-à-vis de moyens d'injection dits "globaux" (affectant l'ensemble du circuit). Cependant, les fautes induites par des impulsions électromagnétiques sont décrites comme étant localisées au voisinage de l'antenne d'injection.

Ce chapitre a pour objectif d'étudier la vulnérabilité du détecteur DVCT vis-à-vis des attaques électromagnétiques. Dans un premier temps, le détecteur sera réglé

de façon à être efficace contre les moyens d'injection "globaux". Ensuite le circuit sera soumis à des impulsions électromagnétiques qui sont supposées "locales" et l'efficacité du détecteur sera étudiée. Enfin, l'importance des caractéristiques physiques de l'antenne d'injection sera abordée.

4.1 Réglage du détecteur

Dans le cas où le circuit est soumis à des attaques globales, un seul détecteur correctement calibré est suffisant pour protéger l'ensemble du circuit. Dans un premier temps, l'AES et un seul détecteur DVCT ont été implémentés sur le circuit cible.

Pour que le détecteur soit fonctionnel, son délai de garde doit être plus long que le temps le plus critique de l'AES. Nous avons mesuré les temps critiques pour 10 000 couples {Message, Clé secrète} tirés aléatoirement. La figure 4.1 présente les chemins critiques mesurés pour ces couples par augmentation statique de la fréquence.

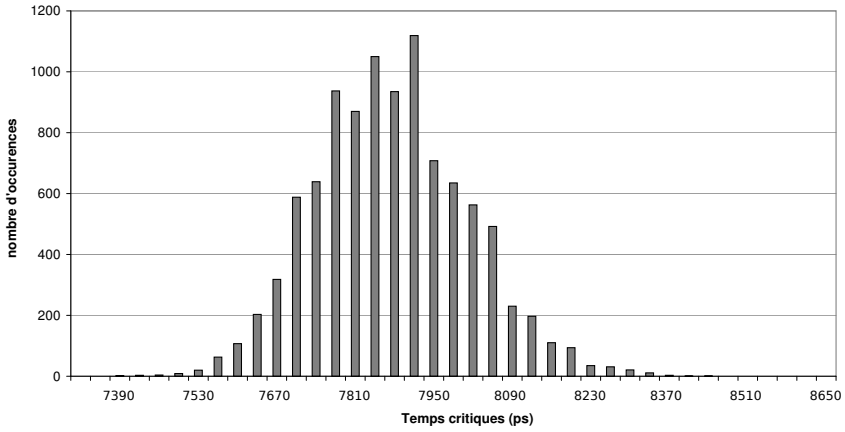

FIGURE 4.1 – Mesure de 10000 chemins critiques de l'AES

La répartition statistique des temps critiques semble suivre une loi normale complètement atténuée après 8,510ns. De ce fait, le délai de garde de la contre-mesure a été calibré pour être supérieur à 8,530ns satisfaisant ainsi aux critères de fonctionnement du détecteur.

Ensuite, le circuit cible a été soumis à des glitchs de tension et d'horloge visant la ronde 9 de l'AES pour 10 000 couples {Message, Clé secrète}. Les figures 4.2 et

4.3 reportent les résultats obtenus dans le cadre d'attaques statiques pour un seul couple {Message, Clé secrète}. Le taux de déclenchement de l'alarme (en rouge) et le taux d'injection de fautes (en bleu) sont représentés en fonction du stress statique appliqué au circuit. La figure 4.2 représente les résultats pour une attaque en fréquence et la figure 4.3 représente les résultats pour une attaque en tension pour un couple {Message, Clé secrète}.

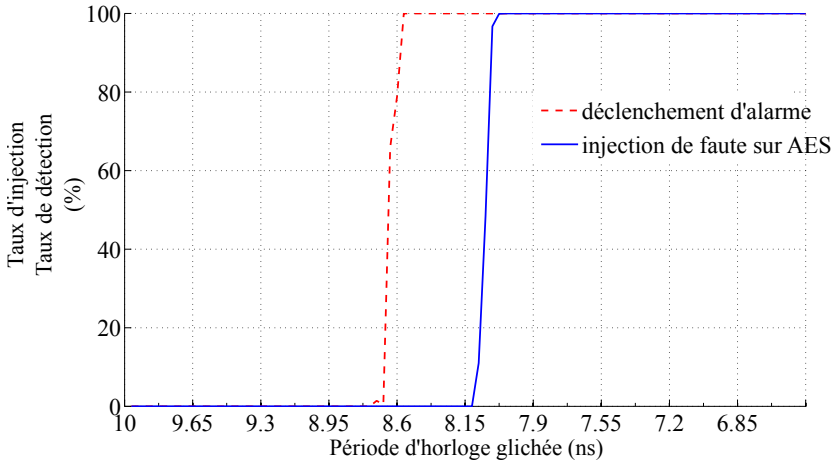

FIGURE 4.2 – Détection de glitchs d'horloge pour un couple {Message, Clé secrète}

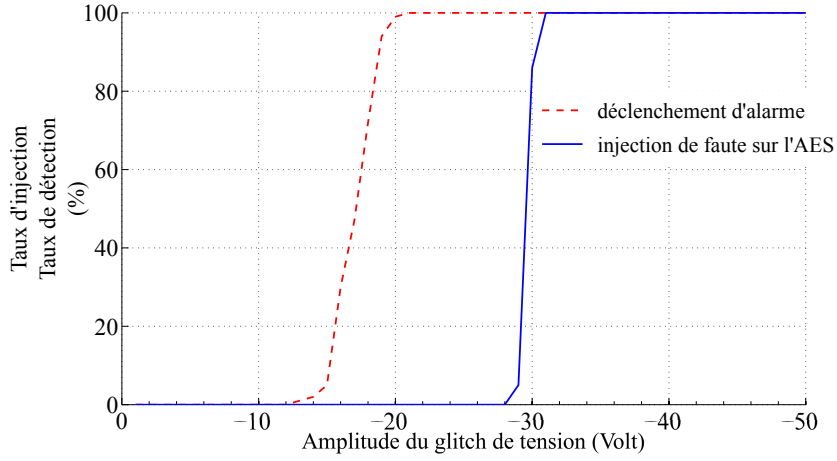

FIGURE 4.3 – Détection de glitchs d'alimentation pour un couple {Message, Clé secrète}

La figure 4.2, illustre une détection systématique pour toutes les périodes d'horloge inférieures 8,55ns et les premières injections de fautes pour des périodes d'horloge inférieure à 8,11ns.

La figure 4.3, illustre une détection systématique pour toutes les amplitudes de glitch supérieures à 20V et les premières injections de fautes pour les amplitudes de glitch supérieures à 28V.

Pour tous les couples {Message, Clé secrète} considérés le détecteur a été efficace et aucune faute n'a été injectée sans que l'alarme ne soit active.

4.2 Efficacité du détecteur vis-à-vis d'attaques électromagnétiques

Dans [Dehbaoui 2012b] les auteurs suggèrent que les impulsions électromagnétique ont pour effet d'injecter des variations de tension confinées au voisinage de l'antenne. Ce caractère de localisation de l'injection nous laisse penser que des fautes peuvent être injectées potentiellement assez loin du détecteur pour ne pas déclencher d'alarme.

4.2.0.1 Analyse des limitations spatiales du détecteur vis-à-vis d'attaques électromagnétiques

Afin d'étudier le comportement du circuit vis-à-vis des injections électromagnétiques, l'intégralité de la surface du circuit non décapsulé a été scannée (XY) par pas de 100μm (position Z au dessus du circuit constante). Pour chacune des positions de l'antenne (1mm de diamètre), l'amplitude du glitch de tension envoyé a été augmentée progressivement de 0V à 200V par pas de 10V. Pour chacune de ces amplitudes, l'attaque a été rejouée 100 fois consécutives. L'état du détecteur et le message chiffré en sortie de l'AES sont relevés. Les figures 4.4 et 4.5 présentent les taux de déclenchement de l'alarme du détecteur en fonction de la position de l'antenne pour deux tensions d'injection différentes. La figure 4.4 a été obtenue en injectant des impulsions de tension de 20ns et de 100V d'amplitude dans l'antenne. La figure 4.5 a été obtenue en injectant des impulsions de tension de 20ns et de 200V d'amplitude dans l'antenne.

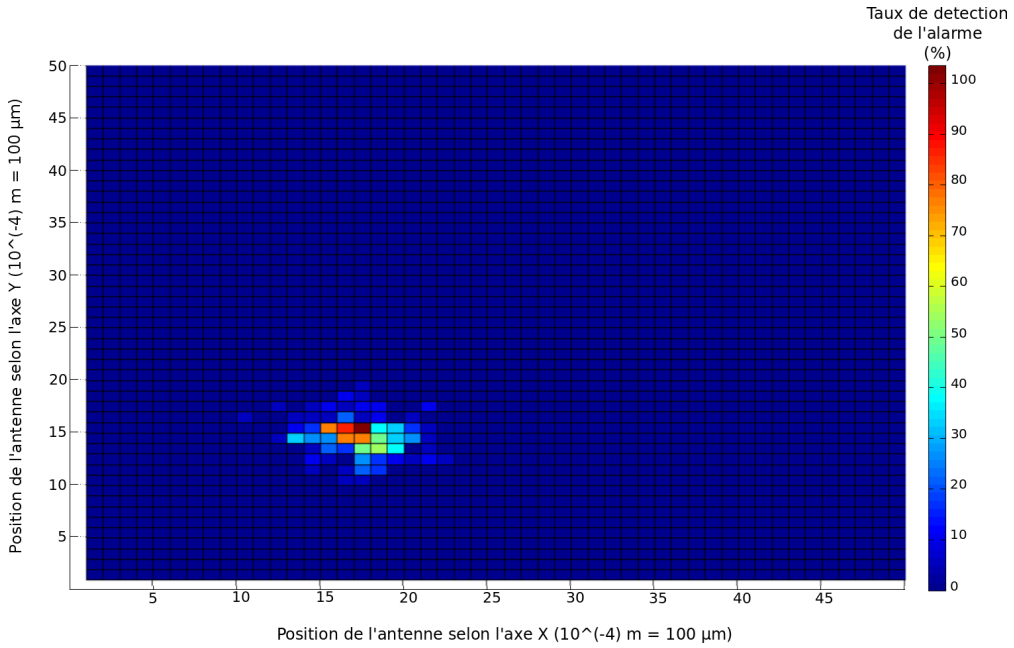

FIGURE 4.4 – Taux de détection de l'alarme en fonction de la position de l'antenne au dessus de la puce : (20ns - 100V)

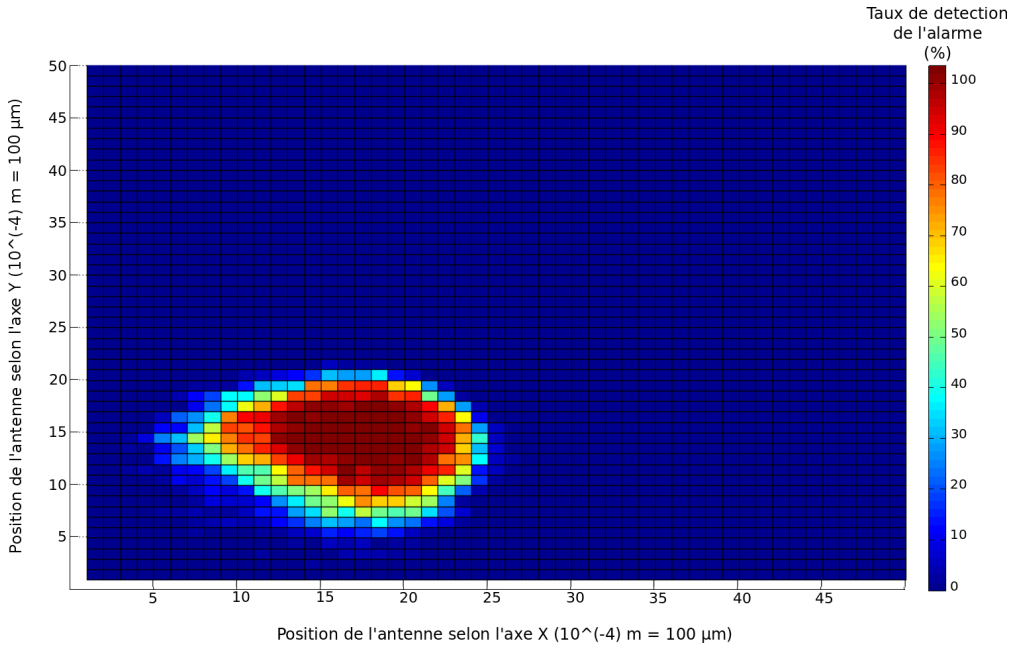

FIGURE 4.5 – Taux de détection de l'alarme en fonction de la position de l'antenne au dessus de la puce : (20ns - 200V)

La zone de couverture du détecteur s'étend avec l'augmentation de la puissance de l'impulsion électromagnétique : de 500μm $\times$ 200μm à 1500μm $\times$ 1500μm. Si la perturbation injectée est plus intense, alors elle a une influence significative sur une plus grande zone du circuit. Dans ce cas, même si l'antenne est éloignée de l'implémentation matérielle du détecteur, les effets de l'impulsion peuvent entraîner un déclenchement de l'alarme. En réduisant l'amplitude de l'injection électromagnétique la zone de détection diminue (mais la sensibilité aux fautes aussi).

Avec les paramètres d'injections considérés (durée de 20ns et impulsions électriques comprises entre 0 et 200V), aucune faute n'a été injectée sur l'AES. Aussi, l'efficacité du détecteur n'est pas remise en cause. Cependant, les résultats obtenus ont montré le caractère local de l'injection électromagnétique. De plus, si des alarmes ont été déclenchées c'est que le délai de garde du détecteur a été modifié, ce qui tend à confirmer que les impulsions électromagnétiques pourraient induire des violations de contraintes temporelles.

4.2.0.2 Injection de fautes non-détectées à l'aide d'impulsions électromagnétiques

Dans la suite de l'étude la tension d'alimentation statique a été réduite à 1,1V (au lieu de 1,2V) afin de rendre les temps critiques de l'AES plus longs et donc plus vulnérables. Il s'agit d'une attaque par diminution statique de la tension. Cette réduction de la tension d'alimentation a aussi pour effet de rendre le délai de garde du détecteur plus long et donc le déclenchement plus sensible. Ensuite, le protocole d'injection électromagnétique présenté précédemment a été utilisé.

Les figures 4.6 et 4.7 présentent les résultats obtenus pour des injections électromagnétiques synchronisées sur la ronde 9 de l'AES et pour un couple {Message, Clé secrète} tiré au hasard avec les paramètres d'injection suivants :

- Amplitude du glitch de tension envoyé dans l'antenne d'injection : 150V
- Durée du glitch de tension envoyé dans l'antenne d'injection : 20ns
- Tension statique d'alimentation du circuit : 1,1V

La figure 4.6 représente le taux de déclenchement de l'alarme en fonction de la position de l'antenne au dessus du boîtier. La diminution de la tension statique a pour effet une augmentation de la zone de déclenchement du détecteur qui couvre cette fois une surface de 3500μm $\times$ 2500μm. La figure 4.7 représente le taux d'injection de fautes sur l'AES pour différentes positions de l'antenne. Nous pouvons observer qu'une faute a été injectée sur une petite zone (200μm $\times$ 200μm) sans que

le détecteur ne se soit déclenché (autour des coordonnées X=38 Y=25).

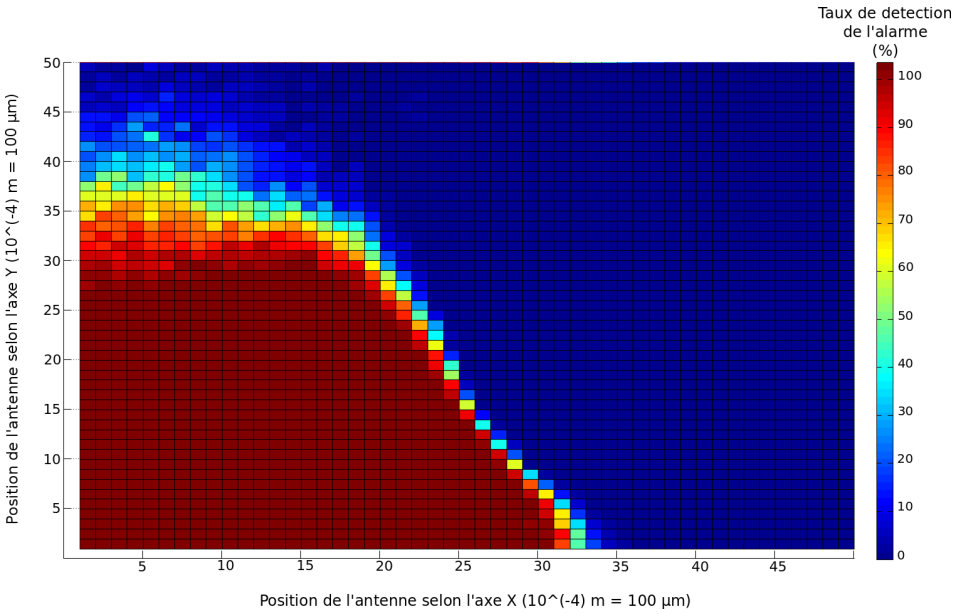

FIGURE 4.6 – Taux de détection de l'alarme en fonction de la position de l'antenne au dessus de la puce : (20ns - 150V - Vdd = 1,1V)

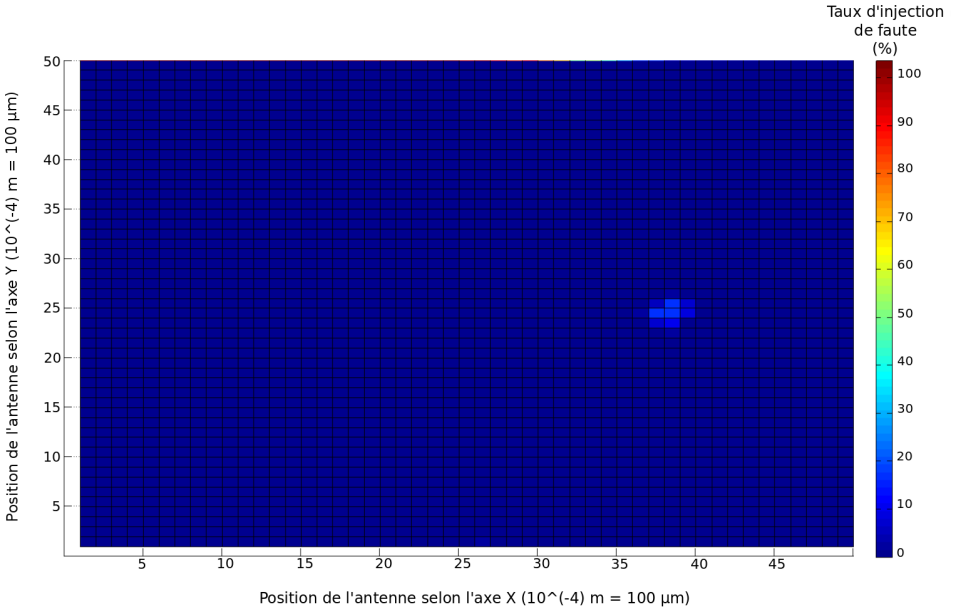

FIGURE 4.7 – Taux d'injection de fautes en fonction de la position de l'antenne au dessus de la puce : (20ns - 150V - Vdd = 1,1V)

Ces expériences montrent qu'une faute a été injectée avec un taux d'injection maximal de 20% sur une zone non protégée par le détecteur. Cette fois l'efficacité du détecteur vis-à-vis de la précision spatiale des injections électromagnétiques est remise en question. De plus, le comportement non déterministe de l'injection de faute tend une nouvelle fois à confirmer l'hypothèse d'injection par violations de contraintes temporelles.

4.2.0.3 Analyse du mécanisme d'injection de fautes lié aux impulsions électromagnétiques

Le détecteur, basé sur un délai de garde, a une efficacité spatialement localisée contre les attaques électromagnétiques. De plus un phénomène de non déterminisme similaire à celui observé pour des injections de fautes par violation de contraintes temporelles sur le temps de setup a été reproduit. Ces résultats tendent à confirmer l'hypothèse que les impulsions électromagnétiques induisent aussi des violations de contraintes temporelles.

Pour vérifier cette hypothèse, de nouvelles injections électromagnétiques ont été conduites afin de retrouver les mêmes fautes que celles obtenues par glitchs de tension et glitchs d'horloge.

Le même protocole que celui décrit précédemment a été utilisé avec, cette fois, l'amplitude du glitch de tension envoyé dans l'antenne d'injection fixée à 190V. Les résultats ont été traités sans tenir compte du déclenchement ou non de l'alarme.

La figure 4.8 représente la zone d'injection de fautes sur le chemin critique de la ronde 9 de l'AES (première faute injectée par glitch d'horloge) pour le même couple {Message, Clé secrète} que celui utilisé dans la section 4.2.0.2.

D'une part, nous avons observé que des fautes injectées à l'aide de glitchs d'horloge peuvent être retrouvées. D'autre part, le détecteur DVCT conçu pour détecter des violations de contraintes temporelles est efficace localement vis-à-vis des injections électromagnétiques. Aussi, nous en avons conclu que les impulsions électromagnétiques peuvent induire localement des violations de contraintes temporelles sur le temps de setup. Cette conclusion n'exclue pas que d'autres mécanismes d'injection puissent intervenir dans d'autres circonstances comme par exemple le basculement d'état d'un registre.

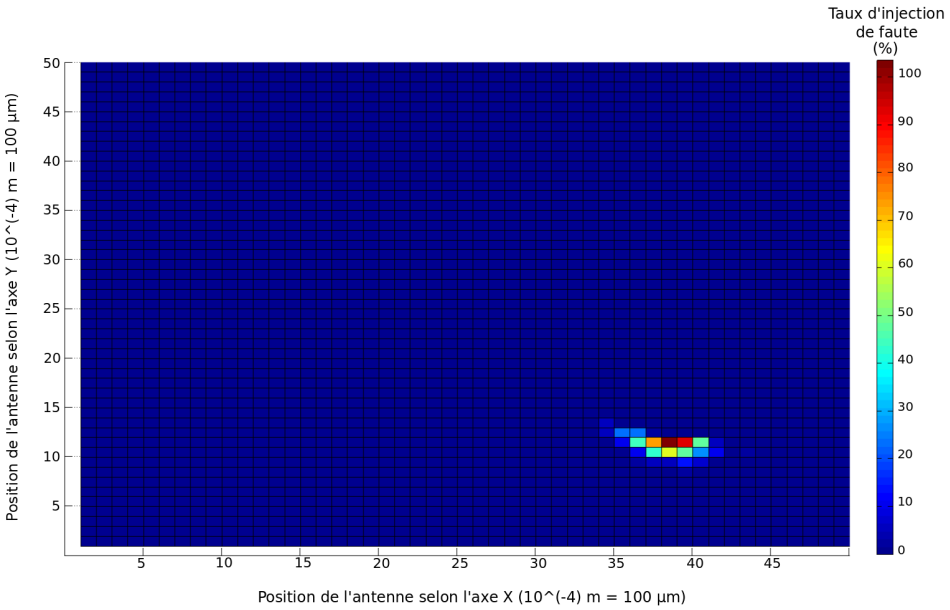

FIGURE 4.8 – Taux d'injection de fautes sur le chemin critique en fonction de la position de l'antenne au dessus de la puce : (20ns - 190V)

4.3 Augmentation de la surface de la zone de protection

Afin d'étendre la surface du circuit protégée contre les violations de contraintes temporelles plusieurs approches ont été proposées :

- Augmentation du délai de garde et de la période de fonctionnement nominale,
- Duplication du nombre de détecteurs.

4.3.1 Augmentation du délai de garde et de la période de fonctionnement nominale

Augmenter le délai de garde au plus proche de la période nominale d'horloge permettrait d'augmenter la sensibilité du détecteur aux petites variations de tension. De plus, augmenter la marge temporelle entre les temps critique de l'AES et la période de fonctionnement nominale rend l'AES moins sensible à l'injection de perturbations. De cette façon, si une impulsion électromagnétique est assez importante pour perturber les calculs de l'AES, ses effets se propageront sur une plus grande partie du circuit et auront une plus grande chance de déclencher le détecteur.

Plus la marge de sensibilité (T_{marge}) entre le délai de garde du détecteur (D_{garde}) et les chemins critiques de l'AES (D_{pMax}) sera grande, plus l'injection de fautes non-

détectées sera limitée. Aussi l'équation à respecter lors du paramétrage du détecteur était la suivante :

$$T_{clk} - T_{setup} + T_{skew} > D_{clk2q} + D_{garde} > D_{clk2q} + D_{pMax} \tag{4.1}$$

Et cette équation devient :

$$T_{clk} - T_{setup} + T_{skew} > D_{clk2q} + D_{garde} > D_{clk2q} + D_{pMax} + T_{marge} \tag{4.2}$$

Cette méthode présente cependant le désavantage d'augmenter le temps de "slack", T_{slack} (temps entre le chemin le plus critique de l'AES et la période d'horloge). Généralement, ce temps T_{slack} est diminué le plus possible pour réduire les temps de calcul au maximum. Le détecteur peut alors introduire un coût en termes de rapidité de fonctionnement du circuit.

4.3.2 Duplication du nombre de détecteurs

Une autre piste pour étendre la zone de protection est de dupliquer un certain nombre de fois le détecteur. De cette façon, chaque détecteur protège une zone différente du circuit. Cette méthode quant à elle induit un surcoût surfacique de l'ordre de 0,25% par détecteur ajouté.

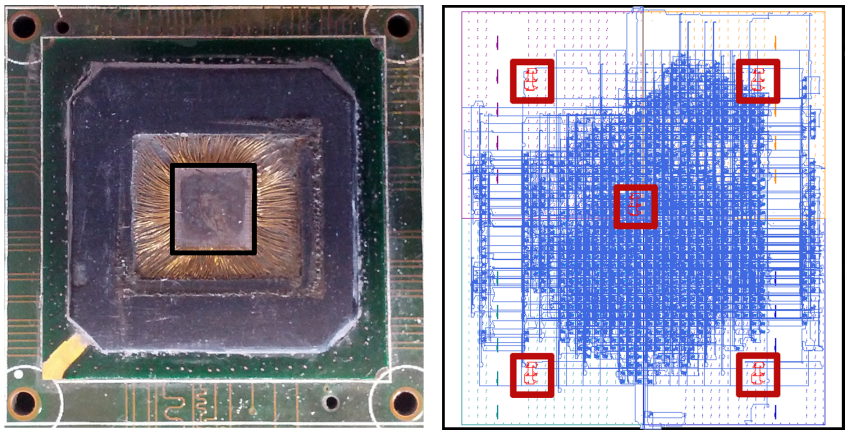

FIGURE 4.9 – (a) FPGA décapsulé pour localisation de la puce - (b) implémentations de l'AES (bleu) et des détecteurs (carrés rouges)

Dans le cadre de cette étude, nous avons choisi d'implémenter 5 détecteurs identiques sur le FPGA cible pour augmenter la zone de détection. La figure 4.9-a est

une image de la face active du circuit cible décapsulé chimiquement. La figure 4.9-b représente le placement et le routage de l'AES ainsi que la position des les 5 détecteurs (carrés rouges) sur le FPGA.

4.4 Influence du choix de l'antenne sur l'aire d'effet d'une attaque électromagnétique

L'augmentation du nombre de détecteurs permet d'augmenter la zone protégée pour une antenne donnée. Cependant la zone d'efficacité d'un détecteur dépendrait également du couplage électromagnétique avec le circuit et donc des paramètres physiques de l'antenne d'injection utilisée. Pour vérifier cette dépendance, des injections ont été réalisées à l'aide de 2 antennes propriétaires différentes présentées figure 4.10. Ces antennes sont constituées d'un noyau en ferrite autour duquel un fil de cuivre est enroulé (4 à 5 spires). L'antenne notée 'G' a un diamètre de $3000\mu m$ et un bout plat. La seconde antenne notée 'F' a un diamètre de $500\mu m$ et un bout appointé de $300\mu m$ suivant les recommandations de [Omarouayache 2013].

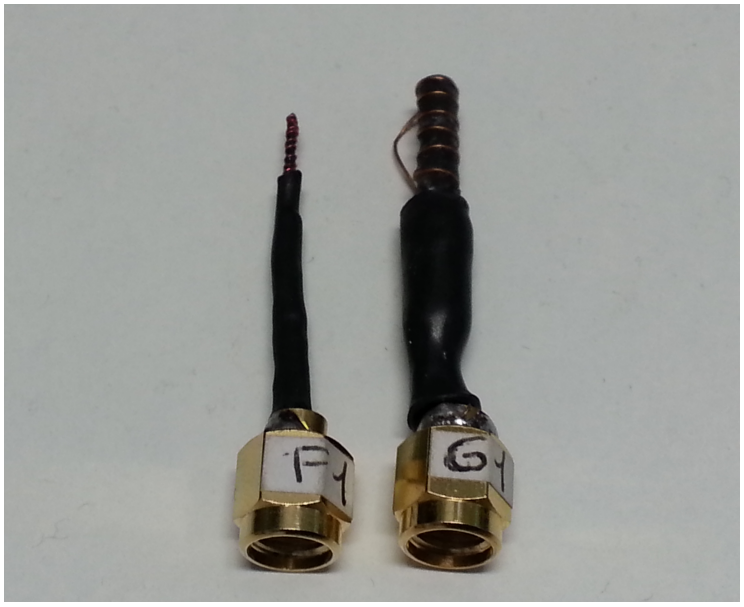

FIGURE 4.10 – Antennes d'injections électromagnétiques propriétaires

Un couple {Message, Clé secrète} a été choisi au hasard et l'implémentation de l'AES protégée par 5 détecteurs a été attaquée en suivant le même protocole que celui décrit précédemment en utilisant cette fois les deux antennes F et G. Le

tableau 4.1 reporte les résultats obtenus avec les deux antennes pour 3 tensions de glitch (75V, 100V et 200V). Chaque ligne du tableau représente 25530 injections : 1702 positions et 15 injections par position.

TABLE 4.1 – Caractéristiques des injections

Antenne	Tension	Taux de détection des injections	Nb fautes	Nb fautes non-détectées
G	200V	32%	1995	0
G	100V	21%	1052	8
F	100V	6%	133	12
F	75V	5,1%	115	5

En utilisant l'antenne 'G' et une tension maximale (200V), 1995 fautes ont été injectées. Cependant, toutes ces injections de fautes ont été accompagnées d'un déclenchement de l'alarme. En diminuant la tension d'injection (100V) moins de fautes sont injectées mais dans le même temps moins d'attaques sont détectées. Dans le cas considéré, 8 fautes ont pu être injectées sans déclenchement de la contre-mesure.

En changeant d'antenne pour une version plus fine 'F' et dont le champ magnétique est concentré via le noyau en ferrite appointé, le nombre de fautes diminue encore puisque les perturbations électromagnétiques affectent de plus petites zones du circuit. Du fait de cette plus grande précision spatiale d'injection, le taux d'injection de fautes non détectées augmente significativement : pour une tension envoyée dans l'antenne de 100V le taux d'injection de fautes non détectées passe de 0,8% avec l'antenne 'G' à 9% avec l'antenne 'F'. Il apparaît alors sur cet exemple que le nombre de détecteurs nécessaires pour protéger un circuit contre des perturbations électromagnétiques dépend de l'antenne d'injection considérée. Ce qui rend le critère de robustesse d'une telle protection très relative à la fois à la cible et au choix de l'antenne.

4.5 Conclusion

Dans ce chapitre un détecteur d'injection de fautes par violations de contraintes temporelles a été implémenté et son efficacité vis-à-vis d'impulsions électromagnétiques a été étudiée. Des injections électromagnétiques réalisées pour différentes

positions d'antenne au dessus du boîtier ont d'abord montré le caractère localisé des injections électromagnétiques et la limitation spatiale du détecteur. Ensuite, en combinant des injections électromagnétiques et une diminution statique de la tension, des fautes non détectées ont été injectées.

L'efficacité locale du détecteur et le comportement non déterministe observé lors des injections de fautes nous ont permis de conclure que les impulsions électromagnétiques entraînent des violations de contraintes temporelles sur le temps de setup.

Pour améliorer la protection fournie par le détecteur deux méthodes ont été proposées.

- La première consiste à augmenter la marge temporelle entre le délai de garde et le chemin le plus critique de l'AES pour contraindre l'attaquant à augmenter la puissance des impulsions pouvant entraîner des injections de fautes. Ce qui a pour conséquence d'augmenter la probabilité que ces injections soient détectées.
- La seconde est de multiplier le nombre de détecteurs pour augmenter le nombre de zones protégées contre les injections électromagnétiques.

Seule la seconde méthode basée sur la duplication du détecteur a été mise en place et son efficacité a été discutée. En effet, si une matrice de détecteurs protège le circuit pour une antenne donnée, il est possible que des fautes non détectées soient injectées à l'aide d'autres antennes construites différemment qui auraient des propriétés d'injection différentes (une meilleure précision spatiale notamment).

Malgré les limitations spatiales du détecteur vis-à-vis des injections électromagnétiques, une forte densité de détecteurs autour des blocs à sécuriser peut s'avérer efficace. De plus, le détecteur a été conçu de telle sorte que son seuil de déclenchement ne dépende pas des données manipulées par le bloc qu'il protège. Cette indépendance entre le détecteur et le reste du circuit n'est peut-être pas vérifiée, a fortiori si les implémentations matérielles de ces deux éléments sont proches.

Dans le prochain chapitre, nous proposons la mise en évidence d'un nouveau canal auxiliaire en présence du détecteur DVCT.

Chapitre 5

Mise en évidence d'un nouveau canal auxiliaire en présence du détecteur DVCT

Sommaire

Pour protéger les circuits des attaques en fautes des détecteurs mesurent des informations relatives à l'état du système (éclairement, tension d'alimentation, fréquence ou température, par exemple). Les limitations de tels détecteurs ont déjà été abordées vis-à-vis des attaques électromagnétiques dans le chapitre 4. Il a été montré que des fautes pouvaient, dans une certaines mesure, être injectées sur le circuit sans activer le détecteur. La question abordée dans ce chapitre concerne les nouvelles vulnérabilités induites par le détecteur lui-même.

La sensibilité (seuil de déclenchement) du détecteur est logiquement indépendante des données traitées par l'AES : elle ne dépend que de la valeur du délai de garde. En effet, son seuil de déclenchement a été conçu pour être constant et ne pas dépendre des données sensibles du circuit. Un tel détecteur devrait théoriquement protéger le circuit d'attaques de type FSA. Cependant, le détecteur doit être implémenté physiquement proche de l'AES pour détecter les injections locales telles que les impulsions électromagnétiques. Cette proximité matérielle peut alors induire un

couplage physique (via la réseau d'alimentation notamment) entre le détecteur et le circuit qu'il protège. En effet, les calculs internes et les changement d'état des registres entraînent des appels de courant perturbant l'environnement électrique du détecteur [Shang 2002], [Zick 2013]. Ces perturbations peuvent avoir une influence suffisante sur le seuil de déclenchement de ce détecteur pour qu'il puisse être corrélé aux données manipulées par l'AES.

Dans ce chapitre, la fuite d'information due au couplage électrique existant entre la contre-mesure et l'implémentation de l'AES-128 est étudié.

5.1 Mise en évidence d'une corrélation entre le détecteur et l'AES

Pour déclencher l'alarme du détecteur plusieurs moyens d'injection de fautes peuvent être utilisés : augmentation de la fréquence, diminution de la tension, impulsions électromagnétiques, etc. Dans le cadre de cette étude, seules des augmentations dynamiques de la fréquence seront traitées pour illustrer cette attaque.

Le stress a été augmenté par pas de 35ps de façon à faire varier une période d'horloge de 10ns (sa valeur nominale) à 7,9ns ($\Delta t = 60 \times 35ps$). Ensuite pour chaque stress $s \in [0..60]$ visant la ronde 1 de l'AES, chaque message M_n a été chiffré m_{max} fois. Pour chaque chiffrement $m \in [1..m_{max}]$ du message M_n avec un stress s, l'attaquant enregistre l'état de l'alarme $A[n, m, s]$ (activée ou désactivée). À la fin de l'expérience, une matrice des états de l'alarme $A[0..n_{max}, 1..m_{max}, 0..s_{max}]$ est obtenue avec dans notre cas $n_{max} = 255$, $m_{max} = 1000$ et $s_{max} = 60$. Cette matrice indique pour un stress, s, imposé au circuit pendant le m^{eme} chiffrement du n^{eme} message si le détecteur a été déclenché ou non. (A[message,itération,intensité du stress] $\in [0, 1]$).

La source d'information que nous voulons exploiter dans cette attaque est le taux de détection du détecteur pour un stress donné en fonction du message en entrée de l'AES. Le taux de déclenchement du détecteur pour les m_{max} chiffrements du n^{eme} message soumis à un stress s est donné par l'équation 5.1 :

$$T_n[s] = \frac{\sum_{i=1}^{m_{max}} A[n, i, s]}{m_{max}} \tag{5.1}$$

La figure 5.1 représente le taux de détection du détecteur en fonction du stress appliqué au circuit pour 3 messages M_n différents ($n = 120$, $n = 139$ et $n = 169$). Bien que le seuil de détection du détecteur ait été conçu pour ne pas dépendre des

données, des variations du seuil de détection sont observées pour les 3 messages considérés dans cet exemple :

- $T_{120}[50] \approx 60\%$,
- $T_{139}[50] \approx 50\%$,
- $T_{169}[50] \approx 30\%$.

Il s'agit ici d'une preuve expérimentale que les données traitées par l'AES ont une influence sur la sensibilité du détecteur.

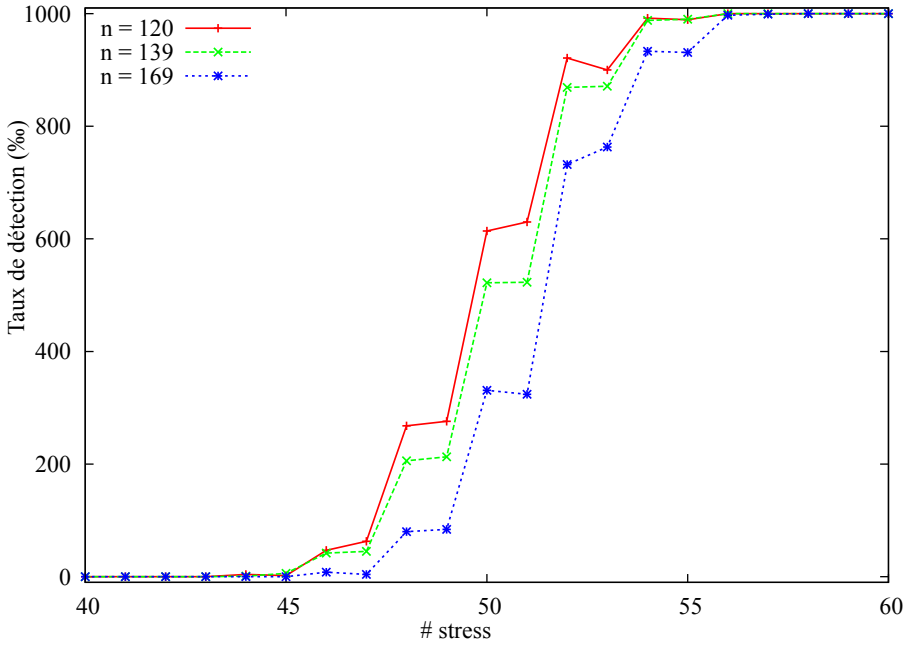

FIGURE 5.1 – Sensibilité de la contre-mesure à la ronde 1 de l'AES pour 3 messages différents

5.2 Exploitation du nouveau canal auxiliaire

L'influence des données traitées par l'AES sur le seuil de déclenchement a été montré expérimentalement. Dans cette section, les techniques de corrélation classiques utilisées dans les attaques de type CPA ou FSA vont être mises en œuvre pour vérifier que l'influence observée figure 5.1 constitue un canal auxiliaire exploitable.

5.2.1 Description de l'attaque

Dans le chemin d'attaque considéré, l'attaque se base sur un effet dérivé de la consommation de courant plutôt que sur la consommation de courant elle-même.

Cette attaque peut être assimilée à une FSA basée sur le seuil de détection du détecteur plutôt que sur le seuil d'injection de fautes.

L'hypothèse est la suivante :

- Les calculs de l'AES ont un effet sur la consommation de courant,
- La consommation de courant a un effet sur le délai de garde du détecteur,
- Les variations du délai de garde peuvent être mesurées en augmentant le stress progressivement jusqu'au déclenchement de l'alarme.

Ensuite, un protocole identique à celui utilisé pour une FSA ou une CPA peut être appliqué pour observer la corrélation entre le seuil de détection du détecteur et les données sensibles de l'AES.

Pour mettre en place l'attaque, la première ronde de l'AES a été visée de façon à ce que les calculs des 16 octets soient encore indépendants. De fait, chaque octet peut être étudié séparément des autres (diviser pour régner). Aussi dans la suite de cette étude seuls les résultats obtenus pour le premier octet seront présentés en détail.

L'objectif est de retrouver la clé secrète K de l'AES (inconnue de l'attaquant).

Pour cela l'attaquant peut commander à l'AES de chiffrer des messages M_n autant de fois qu'il le souhaite (connu de l'attaquant).

L'attaquant va ensuite mesurer le taux de déclenchement du détecteur pour les différents messages considérés quand le circuit est soumis à différents stress $s \in [0..s_{max}]$ (pour $s = 0$ le circuit fonctionne normalement).

$T_{0..n_{max}}[0..s_{max}]$ est le taux de déclenchement du détecteur pour les chiffrements du message M_n avec la clé K quand le circuit est soumis à un stress s : observable par l'attaquant.

Pour que l'attaque puisse être mise en place l'attaquant doit pouvoir comparer ses observations à des grandeurs calculées théoriquement en fonction d'hypothèses faites sur la clé : la *fonction de sélection*. Dans le cas considéré, l'existence d'une relation linéaire entre la valeur d'un bit, b, en sortie du premier SubBytes et le taux de déclenchement du détecteur est supposée.

$SB(b)$ est la valeur du bit b de l'octet considéré en sortie du premier SubBytes de l'AES : inconnu de l'attaquant mais liée physiquement au taux de déclenchement du détecteur. Le i^{eme} octet de SB dépend uniquement du i^{eme} octet de M_n et du i^{eme} octet de K.

L'équation 5.2 explicite l'hypothèse de corrélation linéaire faite par l'attaquant :

$$\exists s \in [1..s_{max}] \text{ tel que } \forall M_n \textbf{ avec } n \in [1..n_{max}], T_{0..n_{max}}[s] = \alpha \times SB[n, K](b) + \beta \tag{5.2}$$

L'attaquant simule ensuite l'algorithme en utilisant les mêmes messages M_n et en faisant des hypothèses sur la clé secrète K_h. Il obtient ainsi des valeurs hypothétiques en sortie du premier SubBytes.

- $SB[0..n_{max}, 0..h_{max}](b)$ sont les hypothèses faites par l'attaquant.

Enfin, l'attaquant cherche la bonne hypothèse de clé en comparant les coefficients de corrélation de Pearson entre les hypothèses de clé et les mesures obtenues pour un stress s donné :

- $\rho_{K_h}[s](SB[0..n_{max}, h](b), T_{0..n_{max}}[s])$ est coefficients de corrélation : calculé par l'attaquant pour validé ou non une hypothèse de clé.

Si l'hypothèse de clé K_h est fausse alors le coefficient de corrélation sera proche de zéro pour tout stress s ($\rho < 0,2$). Par contre, si l'hypothèse de clé est correcte, $K_h = K$, alors le coefficient de corrélation sera différent de zéro pour au moins un stress s ($\rho > 0,2$).

5.2.2 Mise en œuvre de l'attaque

Pour vérifier l'existence d'une corrélation exploitable entre les déclenchements du détecteur (observables) et les données manipulées par l'AES en sortie du premier SubBytes (hypothèses), chaque message a été chiffré 1000 fois pour chacun des 60 stress différents. Ensuite le taux de déclenchement du détecteur, $T_n[s]$, obtenu pour chaque message M_n et pour chaque stress s a été corrélé avec la fonction de sélection $SB[n, h](b)$, pour toutes les hypothèses de clé, K_h.

La figure 5.2 représente le coefficient de corrélation de Pearson, $\rho_{K_h}[s]$, pour toutes les hypothèses de clé, K_h, avec $SB[n, h](1)$ utilisé comme fonction de sélection. Dans ce cas la bonne hypothèse de clé (en rouge) ne ressort pas. C'est à dire qu'il n'existe aucun stress s pour lequel la valeur absolue du coefficient de corrélation relatif à la bonne hypothèse de clé, $\rho_{K_{correct}}[s]$, est très supérieur à tous les autres coefficients $\rho_{K_{incorrect}}[s]$.

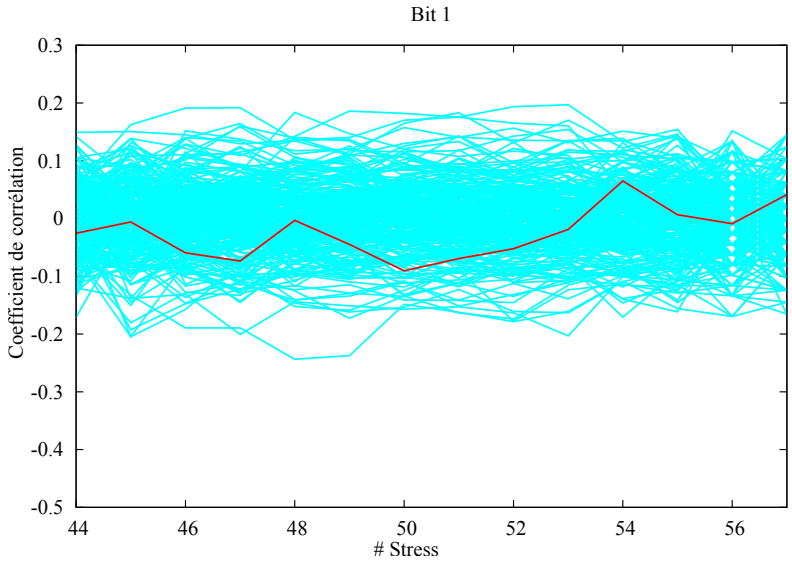

FIGURE 5.2 – Corrélation de Pearson, $\rho_{K_h}[s]$ pour les 256 hypothèses de clé K_h en utilisant la fonction de sélection $SB[n,h](1)$ pour les stress $s \in [44..57]$

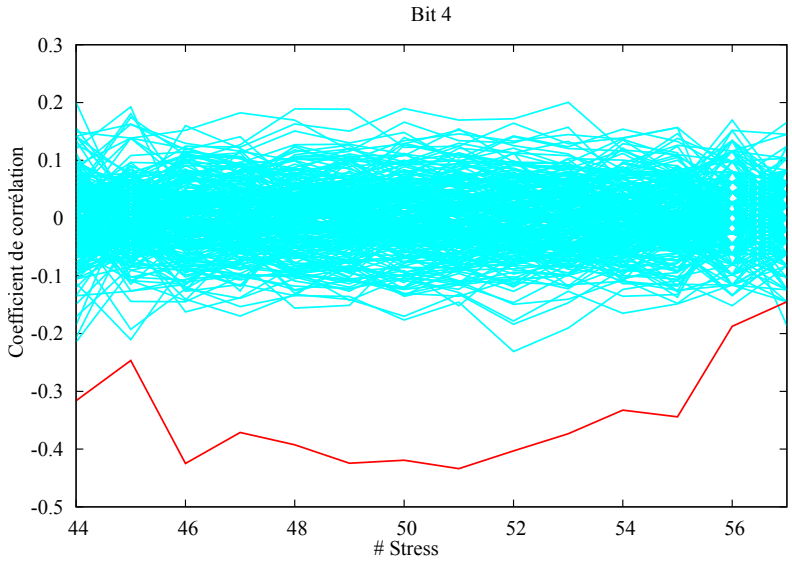

FIGURE 5.3 – Corrélation de Pearson, $\rho_{K_h}[s]$ pour les 256 hypothèses de clé K_h en utilisant la fonction de sélection $SB[n,h](4)$ pour les stress $s \in [44..57]$

La figure 5.3 représente le coefficient de corrélation de Pearson, $\rho_{K_h}[s]$, pour toutes les hypothèses de clé, K_h, avec $SB[n,h](4)$ utilisé comme fonction de sélection. Dans ce cas la bonne hypothèse de clé (en rouge) ressort. C'est à dire qu'il existe au moins un stress s pour lequel la valeur absolue du coefficient de corrélation relatif à la bonne hypothèse de clé, $\rho_{K_{correct}}[s]$, est très supérieur à tous les autres coefficients $\rho_{K_{incorrect}}[s]$.

La figure 5.4 représente le coefficient de corrélation de Pearson, $\rho_{K_h}[s]$, pour toutes les hypothèses de clé, K_h, avec $SB[n,h](8)$ utilisé comme fonction de sélection. Dans ce cas la bonne hypothèse de clé (en rouge) ressort légèrement notamment pour les stress $s = 50$ et $s = 51$ ($\rho > 0,2$).

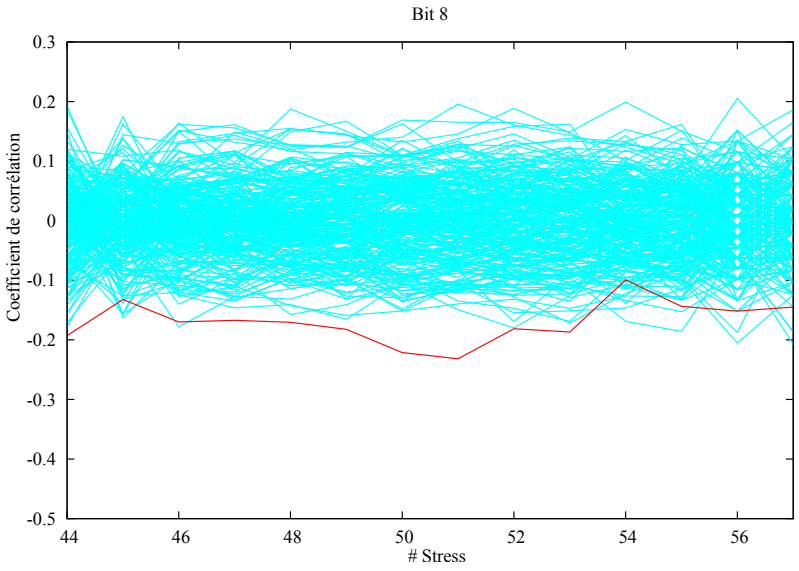

FIGURE 5.4 – Corrélation de Pearson, $\rho_{K_h}[s]$ pour les 256 hypothèses de clé K_h en utilisant la fonction de sélection $SB[n,h](8)$ pour les stress $s \in [44..57]$

Ces résultats confirment que le seuil de déclenchement du détecteur peut être une source d'information exploitable pour retrouver la clé secrète. De plus, ils montrent aussi que le taux de déclenchement n'est pas corrélé à un seul bit en sortie de SUBBYTES mais que tous les bits n'ont pas la même influence sur la sensibilité du détecteur.

Dans le cas décrit ici, l'attaque a été menée avec 15360000 injections de glitchs d'horloge (256 messages × 60 stress × 1000 chiffrements) pour chaque octet. Dans la sous-section suivante le nombre d'injections nécessaires est optimisé en n'appliquant qu'un seul stress s.

5.2.3 Optimisation pratique de l'attaque

La corrélation de Pearson ne peut être efficace que pour des stress mettant le détecteur dans son état non-déterministe. C'est à dire pour des stress s tels que $\forall n \in [0..255], 0 < T_n[s] < 100$. En effet, si le taux de détection est nul quel que soit le message M_n considéré (pour un stress trop faible), il n'y a aucune information à corréler avec la *fonction de sélection*. De la même façon, si le taux de détection est maximal quel que soit le message M_n considéré (pour un stress trop important), il n'y a aucune information à corréler avec la *fonction de sélection*. De plus, n'appliquer au circuit qu'un seul stress bien choisi peut être suffisant pour mener l'attaque.

Les efficacités de différents stress ont été comparées en utilisant la *guessing entropy*, $GE(s)$. La GE est un outil d'évaluation [Standaert 2009] qui indique la position moyenne de la bonne hypothèse de clé après une attaque. Quand l'attaque considérée est une attaque par observation telle que la CPA, la GE est habituellement tracée en fonction du nombre total d'observations. Dans le cas de l'attaque présentée dans ce chapitre qui est une attaque par perturbation, la GE est tracée en fonction du nombre total d'injections. De plus, dans ce cas précis, une alarme doit être déclenchée pour permettre l'attaque et retrouver la clé secrète. Cependant, en fonction de la capacité de l'attaquant à désactiver les effets qu'aurait un déclenchement de l'alarme, ce nombre de détections peut être très limitant. Alors la GE a aussi été tracée en fonction du nombre d'injections ayant induit un déclenchement du détecteur.

Pour obtenir la position moyenne de la bonne hypothèse de clé, l'attaque a été répétée 200 fois et les résultats sont présentés sur les figures 5.5 et 5.6. La figure 5.5 représente la position moyenne de la bonne hypothèse de clé ($GE(s)$) pour 4 stress $s = 45$, $s = 46$, $s = 49$ et $s = 55$ en fonction du nombre total d'injections. Dans cette figure il apparaît que la GE tend plus rapidement vers 1 quand le stress considéré ($48 \leq s \leq 53$) entraîne un taux de détection compris entre 20% et 80%.

Dans le cas présenté dans ce chapitre, pour $s = 49$ il a fallu à peu près 7000 glitchs d'horloge injectés pour retrouver la bonne hypothèse de clé. Cependant le nombre de déclenchements du détecteur peut être une limitation pratique à la mise en place de cette attaque. La figure 5.6 présente les mêmes résultats en fonction du nombre d'alarmes déclenchées. Cette figure montre qu'attaquer le circuit avec un stress induisant un taux de déclenchement inférieur à 30% peut être une stratégie intéressante pour minimiser le nombre de détections nécessaires afin de retrouver la bonne hypothèse de clé.

Dans le cas présenté ici, un stress moyen $s = 49$ minimise le nombre total d'injections (environ 7000) et un stress faible $s = 46$ minimise le nombre total d'alarmes déclenchées (environ 800).

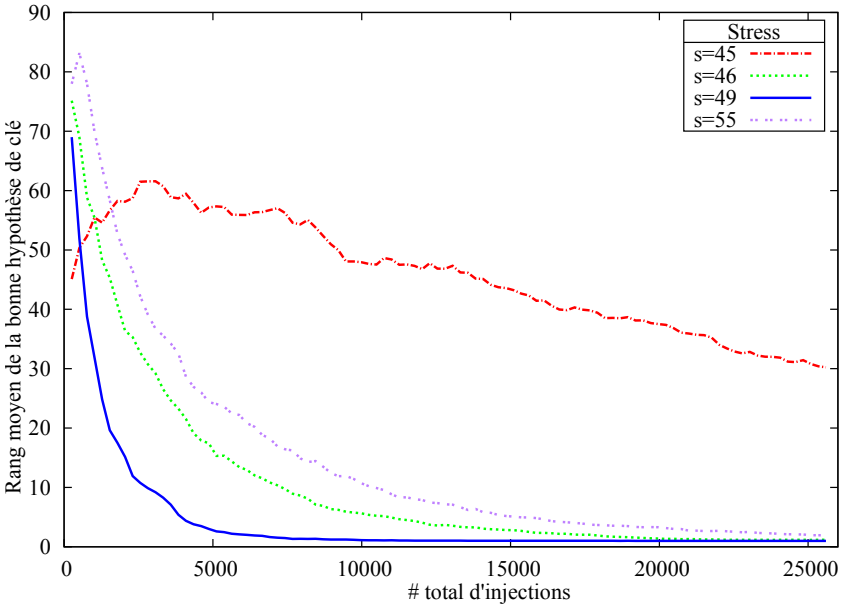

FIGURE 5.5 – Nombre total d'injections

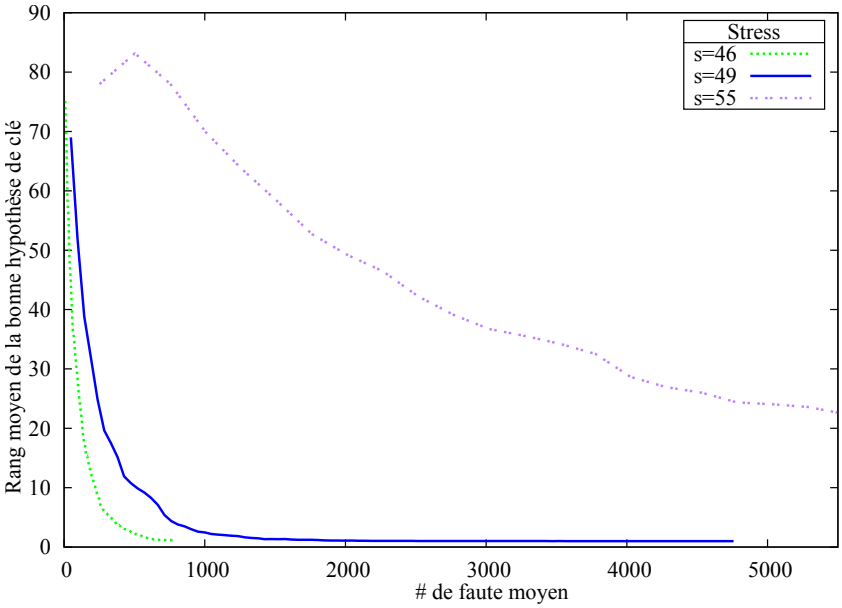

FIGURE 5.6 – Nombre total de déclenchements

5.2.4 Résultats obtenus pour l'ensemble des octets de la clé

La figure 5.7, illustre les résultats obtenus pour chaque octet de la clé secrète avec un stress $s = 49$ et 1000 chiffrements pour chaque message considéré (soit un total de 4081000 injections). Le nombre total de chiffrements pourrait être diminué sans altérer le résultat mais cette section n'a pas pour objectif d'optimiser l'attaque mais de montrer la similarité des résultats pour tous les octets de la clé.

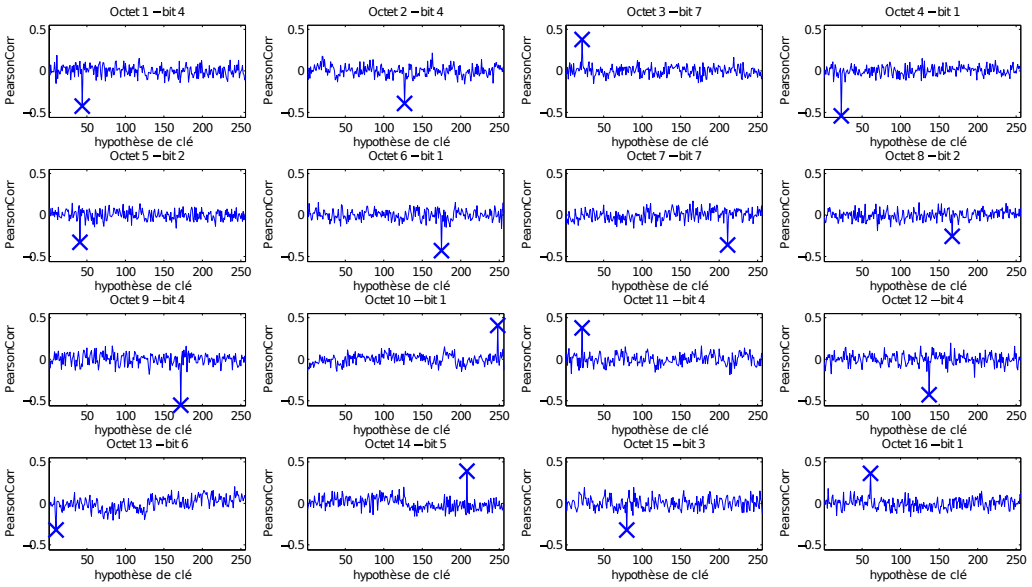

FIGURE 5.7 – Indice de corrélation pour chaque octet de la clé avec $s = 49$ et $m_{max} = 1000$

Sur la figure 5.7, pour chaque octet, seul le bit ayant donné les meilleurs résultats de corrélation est représenté. Il apparaît que le bit ayant la plus grosse influence linéaire sur le seuil de déclenchement du détecteur n'est pas toujours le même (bit 4 pour les octets 1 et 2, bit 7 pour l'octet 3, etc.) et que la corrélation n'est pas toujours positive.

5.3 Conclusion

Dans ce chapitre un nouveau chemin d'attaque a été mis en évidence. Cette attaque exploite un couplage électrique entre blocs logiquement indépendants pour retrouver des informations secrètes. Pour illustrer cette fuite d'informations, un AES

a été implémenté sur un FPGA. Cet AES était protégé contre les injections de fautes par un détecteur basé sur un délai de garde. Ce détecteur et l'AES étant supposés être deux blocs logiquement indépendants, le seuil de déclenchement du détecteur ne devrait pas dépendre des calculs de l'AES.

Cependant, les expériences ont montré qu'une corrélation entre la sensibilité du détecteur et les données sensibles manipulées par l'AES existait. Cette fuite d'information nous a permis de retrouver la clé secrète de l'AES.

Ce résultat est particulièrement intéressant parce qu'il met en avant une vulnérabilité du détecteur vis-à-vis d'une attaque de type FSA alors que celui-ci était conçu pour empêcher ce genre d'attaques.

Néanmoins, ce chemin d'attaque implique que de nombreuses alarmes soient déclenchées, ce qui peut être une limitation à son efficacité pratique. Dans le cadre de cette étude nous avons essayé de diminuer le nombre de détections en attaquant le circuit avec un stress plus faible.

Il faut toutefois préciser que le déclenchement d'une alarme n'est pas systématiquement problématique. En effet, la stratégie de sécurité du circuit peut être d'effacer la clé secrète stockée dans une mémoire non volatile. Cette stratégie entraîne une surconsommation d'énergie significative pouvant être détectée par l'attaquant qui peut alors couper l'alimentation du circuit avant que cet effacement ne soit effectué.

Le canal auxiliaire présenté dans ce chapitre exploite donc un couplage électrique entre blocs indépendants qui pourraient être applicable à d'autres capteurs.

Conclusion générale

L'objectif de cette thèse était d'étudier dans un premier temps les moyens d'injection de fautes pouvant conduire à des violations de contraintes temporelles et permettant la cryptanalyse physique de circuits sécurisés. Dans un second temps, elle portait sur l'étude des vulnérabilités d'un détecteur conçu pour protéger le circuit cible des injections de fautes par violations des contraintes temporelles. En effet, si un algorithme cryptographique peut être mathématiquement sûr, son implémentation matérielle quant à elle peut être la cible de nombreuses attaques et les protections contre certaines d'entre-elles peuvent à leur tour être la source de nouvelles vulnérabilités. La caractérisation du niveau de sécurité d'un circuit devient donc une tache particulièrement difficile.

Dans le premier chapitre, un état de l'art sur les attaques matérielles permettant une cryptanalyse physique des circuits sécurisés a été présenté. Ce chapitre a permis d'une part, d'orienter l'étude vers certains moyens d'injection supposés induire des violations de contraintes temporelles et d'autre part, d'introduire différentes méthodes pour traiter les résultats obtenus dans la cadre d'une cryptanalyse physique.

Dans le chapitre 2, les bancs d'injections utilisés dans le cadre de cette thèse (variations statiques ou dynamiques de la tension, de la fréquence, de la température et de l'environnement électromagnétique) ont été présentés. Les implémentations matérielles d'un AES-128 qui a servi de cible et d'un détecteur dont l'efficacité a été discutée ont aussi été décrits. Pour finir, un voltmètre embarqué qui a été utilisé pour observer les perturbations internes du circuit cible a été introduit.

Dans le chapitre 3, les techniques d'injection de fautes statiques présentées dans l'état de l'art (diminution de la tension, augmentation de la fréquence et augmentation de la température) ont été mises en pratique sur une implémentation de l'AES sur FPGA. Il a été démontré expérimentalement qu'une diminution de la tension d'alimentation ou qu'une augmentation de la température induisent les mêmes fautes dans un circuit synchrone qu'une augmentation de la fréquence de fonctionnement. Aussi nous avons conclu que ces trois moyens d'injection partagent le même mécanisme d'injection à savoir les violations des contraintes de temps de setup. Les évolutions des temps critiques de l'AES en fonction de la tension d'alimentation et de la température ont aussi été observées. Dans les plages de stress considérées, c'est deux paramètres ont des influence quasi-linéaire sur l'évolution des temps de

propagation. Cependant, les attaques statiques ne permettent pas de viser systématiquement une ronde donnée de l'AES. Alors, les perturbations transitoires de tension ont été étudiées et les fautes injectées de cette façon ont été comparées aux fautes obtenues par variations dynamiques de la fréquence. D'abord, une phase d'essais/erreurs a permis de converger vers une double injection (avec deux générateurs de glitchs de tension) et de trouver des paramètres permettant une injection de fautes avec une bonne précision temporelle. Les fautes induites par les diminutions dynamiques de tension étant identiques aux fautes induites par les variations de fréquence, nous avons conclu qu'il s'agissait encore dans ce cas de violations des contraintes de temps de setup. Enfin, l'implémentation d'un voltmètre intégré sur le FPGA cible nous a permis d'observer les perturbations réellement injectées et de mieux comprendre le mécanisme d'injection de fautes. Nous avons constaté notamment qu'un glitch de tension entraîne des oscillations amorties sur la tension d'alimentation du circuit synchronisées sur les fronts montant et descendant du glitch. Il a été démontré expérimentalement que des glitchs positifs pouvaient aussi induire des violations de contraintes de temps de setup. Enfin, les observations faites à l'aide du voltmètre embarqué nous ont aussi permis de conclure a posteriori que les paramètres trouvés empiriquement pour la double injection de glitchs étaient bien adaptés à l'injection de fautes (bonne résolution temporelle).

L'efficacité du détecteur basé sur un délai de garde et protégeant le circuit contre les injections de fautes par violations de contraintes temporelles a été étudiée face à des injections électromagnétiques a été étudié dans le chapitre 4. Les expériences ont montré le caractère localisé des injections électromagnétiques et la limitation spatiale de l'efficacité du détecteur. De plus, nous avons montré que les impulsions électromagnétiques induisent localement des violations de contraintes temporelles. Pour améliorer la protection fournie par le détecteur, deux méthodes ont été proposées. La première est d'augmenter la marge temporelle entre le délai de garde et le chemin le plus critique de l'AES pour contraindre l'attaquant à augmenter la puissance des injections pouvant entraîner une faute et donc la probabilité que ces injections soient détectées. La seconde est de multiplier le nombre de détecteurs pour augmenter le nombre de zones protégées contre les injections électromagnétiques. Seule la seconde méthode a été mise en place et son efficacité a été discutée. En effet, si une matrice de détecteurs protège efficacement le circuit pour une antenne donnée, il se peut que des fautes non détectées puissent être injectées avec d'autres antennes construites différemment de façon à avoir des effets encore plus

localisés.

Finalement, dans le chapitre 5, un nouveau canal auxiliaire exploitant les variations de la sensibilité du détecteur a été proposé et validé expérimentalement. Cette attaque exploite le couplage électrique existant entre les données manipulées par l'AES et la valeur du délai de garde du détecteur. À cause des variations électriques induites par l'AES sur le réseau d'alimentation, la sensibilité de l'alarme varie suffisamment pour que cette corrélation soit exploitable. Ce résultat est particulièrement intéressant parce qu'il met en avant une vulnérabilité du détecteur vis-à-vis d'une attaque de type FSA alors que celui-ci était conçu pour empêcher ce genre d'attaques. Néanmoins, ce chemin d'attaque implique que beaucoup d'alarmes soient déclenchées, ce qui peut être une limitation à son efficacité pratique.

En terme de perspective à ces différents résultats obtenus, le voltmètre embarqué pourrait être utilisé pour observer des attaques électromagnétiques et mieux comprendre le mécanisme de propagation et ainsi améliorer la disposition des détecteurs. Des techniques de masquage pourraient être utilisées pour décorréler les variations de la sensibilité de ce dernier et les données sensibles. Ainsi l'attaque présentée dans le chapitre 5 pourrait être reconduite pour vérifier l'efficacité de cette méthode de protection. Enfin, il est apparu que le seuil de détection d'un détecteur pouvait dépendre de l'implémentation matérielle de l'algorithme. Aussi l'étude de ce seuil de détection pourrait être utilisé dans le domaine de la détection de chevaux de Troie matériels (modifications malveillantes du circuit). En effet la présence de chevaux de Troie peut avoir pour effet d'augmenter la consommation locale de courant et de ce fait modifier les seuils de déclenchement des détecteurs.

Bibliographie

[Agoyan 2010a] M. Agoyan, J.-M. Dutertre, A.-P. Mirbaha, D. Naccache, A.-L. Ribotta et A. Tria. *How to flip a bit ?* In On-Line Testing Symposium (IOLTS), 2010. (Cité en page 13.)

[Agoyan 2010b] M. Agoyan, J.M. Dutertre, D. Naccache, B. Robisson et A. Tria. *When clocks fail : On critical paths and clock faults.* Smart Card Research and Advanced Application, 2010. (Cité en pages 13 et 39.)

[BarEl 2006] H. BarEl, H. Choukri, D. Naccache, M. Tunstall et C. Whelan. *The Sorcerer's Apprentice Guide to Fault Attacks.* In Special Issue on Cryptography and Security, 2006. (Cité en page 34.)

[Barenghi 2010] Alessandro Barenghi, Guido Bertoni, Luca Breveglieri, Mauro Pellicioli et Gerardo Pelosi. *Low Voltage Fault Attacks to AES.* In Hardware-Oriented Security and Trust (HOST), 2010. (Cité en page 13.)

[Barenghi 2012] A. Barenghi, L. Breveglieri, I. Koren et D. Naccache. *Fault Injection Attacks on Cryptographic Devices : Theory, Practice, and Countermeasures.* Proceedings of the IEEE, 2012. (Cité en page 34.)

[Bayon 2012] Pierre Bayon, Lilian Bossuet, Alain Aubert, Viktor Fischer, François Poucheret, Bruno Robisson et Philippe Maurine. *Contactless Electromagnetic Active Attack on Ring Oscillator Based True Random Number Generator.* In Constructive Side-Channel Analysis and Secure Design (COSADE), 2012. (Cité en page 34.)

[Biham 1997] E. Biham et A. Shamir. *Differential fault analysis of secret key cryptosystems.* In Advances in Cryptology (CRYPTO), Lecture Notes in Computer Science, 1997. (Cité en pages 18 et 19.)

[Blömer 2003] Johannes Blömer et Jean-Pierre Seifert. *Fault based cryptanalysis of the advanced encryption standard (AES).* In Financial Cryptography. Springer, 2003. (Cité en page 18.)

[Boneh 1997] D. Boneh, R.A. DeMillo et R.J. Lipton. *On the importance of checking cryptographic protocols for faults.* In EUROCRYPT, Lecture Notes in Computer Science, 1997. (Cité en pages 18 et 19.)

[Brier 2004] E. Brier, C. Clavier et F. Olivier. *Correlation Power Analysis with a Leakage Model.* In Cryptographic Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science, 2004. (Cité en page 14.)

[Chari 2003] Suresh Chari, Josyula R Rao et Pankaj Rohatgi. *Template attacks.* In Cryptographic Hardware and Embedded Systems (CHES). Springer, 2003. (Cité en page 14.)

[Cho 2005] Kyoung Youn Cho, Subhasish Mitra et Edward J McCluskey. *Gate exhaustive testing.* In Test Conference (ITC). IEEE, 2005. (Cité en page 34.)

[DeBusschere 2012] E DeBusschere et M McCambridge. *Modern Game Console Exploitation.* 2012. (Cité en page 13.)

[Dehbaoui 2012a] A. Dehbaoui, J.-M. Dutertre, B. Robisson et A. Tria. *Electromagnetic Transient Faults Injection on a Hardware and Software Implementation of AES.* In Fault Diagnosis and Tolerance in Cryptography (FDTC), 2012. (Cité en page 13.)

[Dehbaoui 2012b] Amine Dehbaoui, Jean-Max Dutertre, Bruno Robisson, P Orsatelli, Philippe Maurine et Assia Tria. *Injection of transient faults using electromagnetic pulses-practical results on a cryptographic system.* Rapport technique, IACR Cryptology ePrint Archive, 2012. (Cité en pages 34, 88 et 92.)

[Dehbaoui 2013] Amine Dehbaoui, Amir-Pasha Mirbaha, Nicolas Moro, Jean-Max Dutertre et Assia Tria. *Electromagnetic Glitch on the AES Round Counter.* In Constructive Side-Channel Analysis and Secure Design (COSADE), 2013. (Cité en page 34.)

[Di Natale 2007] Giorgio Di Natale, Marie-Lise Flottes et Bruno Rouzeyre. *On-Line Self-Test of AES Hardware Implementations.* In International Conference on Dependable Systems and Networks. Citeseer, 2007. (Cité en page 18.)

[Diffie 1976] W. Diffie et M. E. Hellman. *New Directions in Cryptography.* In IEEE Transactions on Information Theory, 1976. (Cité en page 9.)

[Djellid-Ouar 2006] A. Djellid-Ouar, G. Cathebras et F. Bancel. *Supply voltage glitches effects on CMOS circuits.* In Design and Test of Integrated Systems in Nanoscale Technology (DTIS), 2006. (Cité en page 34.)

[Dudek 2000] Piotr Dudek, Stanislaw Szczepanski et John V Hatfield. *A high-resolution CMOS time-to-digital converter utilizing a Vernier delay line.* Solid-State Circuits, IEEE Journal of, vol. 35, 2000. (Cité en pages 45 et 46.)

[Dutertre 2009] Jean-Max Dutertre, Assia Tria, Bruno Robisson et Michel Agoyan. *Low cost fault injection method for security characterization.* In E-Smart, 2009. (Cité en page 39.)

[Dutertre 2012] J-M Dutertre, A-P Mirbaha, David Naccache, A-L Ribotta, Assia Tria et Thierry Vaschalde. *Fault round modification analysis of the advanced encryption standard.* In Hardware-Oriented Security and Trust (HOST). IEEE, 2012. (Cité en page 19.)

[Endo 2011] Sho Endo, Takeshi Sugawara, Naofumi Homma, Takafumi Aoki et Akashi Satoh. *An on-chip glitchy-clock generator for testing fault injection attacks.* J. Cryptographic Engineering, 2011. (Cité en page 39.)

[Endo 2012] Sho Endo, Yang Li, Naofumi Homma, Kazuo Sakiyama, Kazuo Ohta et Takafumi Aoki. *An Efficient Countermeasure against Fault Sensitivity Analysis Using Configurable Delay Blocks.* In Fault Diagnosis and Tolerance in Cryptography (FDTC). IEEE, 2012. (Cité en pages 19, 50 et 88.)

[Fukunaga 2009] T. Fukunaga et J. Takahashi. *Practical Fault Attack on a Cryptographic LSI with ISO/IEC 18033-3 Block Ciphers.* In Fault Diagnosis and Tolerance in Cryptography (FDTC), 2009. (Cité en page 39.)

[Gammel 2010] B.M. Gammel et S. Mangard. *On the Duality of Probing and Fault Attacks.* In J.Electron.Test., 2010. (Cité en page 13.)

[Gandolfi 2001] K. Gandolfi, C. Mourtel et F. Olivier. *Electromagnetic analysis : concrete result.* In Cryptographic Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science, 2001. (Cité en pages 12 et 14.)

[Gierlichs 2008] Benedikt Gierlichs, Lejla Batina, Pim Tuyls et Bart Preneel. *Mutual information analysis.* In Cryptographic Hardware and Embedded Systems (CHES). Springer, 2008. (Cité en page 14.)

[Giraud 2005] C. Giraud. *DFA on AES.* In Advanced Encryption Standard – AES, volume 3373 of *Lecture Notes in Computer Science*, 2005. (Cité en pages 19, 20, 21 et 23.)

[Gomina 2014] Kamil Gomina, Jean-Baptiste Rigaud, Philippe Gendrier, Philippe Candelier et Assia Tria. *Power supply glitch attacks : Design and evaluation of detection circuits.* In Hardware-Oriented Security and Trust (HOST). IEEE, 2014. (Cité en page 89.)

[Ha 2011] D. Ha, K. Woo, S. Meninger, T. Xanthopoulos, E. Crain et D. Ham. *Time-Domain CMOS Temperature Sensors With Dual Delay-Locked Loops*

for Microprocessor Thermal Monitoring. Very Large Scale Integration (VLSI) Systems, 2011. (Cité en page 33.)

[Handschuh 1999] H. Handschuh, P. Paillier et J. Stern. *Probing Attacks on Tamper-Resistant Devices.* In Cryptographic Hardware and Embedded Systems (CHES), 1999. (Cité en page 13.)

[Horstmann 1989] J.U. Horstmann, H.W. Eichel et R.L. Coates. *Metastability behavior of CMOS ASIC flip-flops in theory and test.* Solid-State Circuits, IEEE Journal of, 1989. (Cité en page 30.)

[Joye 2007] Marc Joye, Pascal Manet et J-B Rigaud. *Strengthening hardware AES implementations against fault attacks.* IET Information Security, 2007. (Cité en page 18.)

[Kahn 2008] David Kahn. *Codebreakers : L'histoire de l'écriture secrète.* 2008. (Cité en page 8.)

[Kim 2012] Chong Hee Kim. *Improved Differential Fault Analysis on AES Key Schedule.* Information Forensics and Security, IEEE Transactions on, 2012. (Cité en page 19.)

[Kocher 1999] Paul C. Kocher, Joshua Jaffe et Benjamin Jun. *Differential Power Analysis.* In CRYPTO, 1999. (Cité en pages 12, 14, 15 et 17.)

[Koeune 2005] F. Koeune et F.-X. Standaert. *A Tutorial on Physical Security and Side-Channel Attacks.* In Foundations of Security Analysis and Design III, Lecture Notes in Computer Science, 2005. (Cité en page 12.)

[Kömmerling 1999] Oliver Kömmerling et Markus G. Kuhn. *Design Principles for Tamper-Resistant Smartcard Processors.* In Proceedings of the USENIX Workshop on Smartcard Technology, 1999. (Cité en page 13.)

[Lashermes 2012] R. Lashermes, G. Reymond, J.-M. Dutertre, J. Fournier, B. Robisson et A. Tria. *A DFA on AES based on the Entropy of Error Distributions.* In Fault Diagnosis and Tolerance in Cryptography (FDTC), 2012. (Cité en page 19.)

[Li 2012] Yang Li, Kazuo Ohta et Kazuo Sakiyama. *New Fault-Based Side-Channel Attack Using Fault Sensitivity.* IEEE Transactions on Information Forensics and Security, 2012. (Cité en pages 19, 25, 26 et 27.)

[Maistri 2007] Paolo Maistri, Pierre Vanhauwaert et Régis Leveugle. *A novel double-data-rate AES architecture resistant against fault injection.* In Fault Diagnosis and Tolerance in Cryptography, (FDTC). IEEE, 2007. (Cité en page 18.)

[Maistri 2011] Paolo Maistri. *Countermeasures against fault attacks : The good, the bad, and the ugly.* In On-Line Testing Symposium (IOLTS). IEEE, 2011. (Cité en page 18.)

[Mangard 2010] Stefan Mangard, Elisabeth Oswald et Thomas Popp. Power analysis attacks : Revealing the secrets of smart cards. Springer Publishing Company, Incorporated, 2010. (Cité en page 14.)

[Moradi 2006] A. Moradi, M. T. Manzuri Shalmani et M. Salmasizadeh. *A Generalized Method of Differential Fault Attack Against AES Cryptosystem.* In Cryptographic Hardware and Embedded Systems (CHES), 2006. (Cité en page 19.)

[Moro 2013] Nicolas Moro, Amine Dehbaoui, Karine Heydemann, Bruno Robisson et Emmanuelle Encrenaz. *Electromagnetic fault injection : towards a fault model on a 32-bit microcontroller.* In Fault Diagnosis and Tolerance in Cryptography (FDTC). IEEE, 2013. (Cité en page 13.)

[NIST 1999] NIST. *Data Encryption Standard (DES).* Federal Information Processing Standards Publication, n. 46, 3, 1999. (Cité en page 9.)

[NIST 2001] NIST. *Announcing the ADVANCED ENCRYPTION STANDARD (AES).* Federal Information Processing Standards Publication 197, 2001. (Cité en page 9.)

[Omarouayache 2013] R. Omarouayache, J. Raoult, S. Jarrix, L. Chusseau et P. Maurine. *Magnetic Microprobe Design for EM Fault AttackMagnetic Microprobe Design for EM Fault Attack.* In emceurope, 2013. (Cité en page 99.)

[Piret 2003] G. Piret et J.-J. Quisquater. *A Differential Fault Attack Technique against SPN Structures, with Application to the AES and Khazad.* In Cryptographic Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science, 2003. (Cité en pages 19, 23 et 24.)

[Poucheret 2011] François Poucheret, Karim Tobich, M Lisarty, Laurent Chusseau, Bruno Robisson et Philippe Maurine. *Local and direct EM injection of power into CMOS integrated circuits.* In Fault Diagnosis and Tolerance in Cryptography (FDTC). IEEE, 2011. (Cité en page 88.)

[Quisquater 2002] Jean-Jacques Quisquater et David Samyde. *Eddy current for Magnetic Analysis with Active Sensor.* In Esmart, 2002. (Cité en page 34.)

[Razavi 2008] Behzad Razavi. Fundamentals of Microelectronics. Wiley, 2008. (Cité en page 33.)

[Rivest 1978] R. Rivest, A. Shamir et L. Adleman. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.* In Communications of the ACM, 1978. (Cité en page 9.)

[Robisson 2007] Bruno Robisson et Pascal Manet. *Differential Behavioral Analysis.* In Cryptographic Hardware and Embedded Systems (CHES), 2007. (Cité en page 19.)

[Roche 2011] T. Roche, V. Lomné et K. Khalfallah. *Combined Fault and Side-Channel Attack on Protected Implementations of AES.* In Smart Card Research and Advanced Applications, 2011. (Cité en pages 13, 19, 21, 22 et 23.)

[Schlosser 2012] A. Schlosser, D. Nedospasov, J. Kramer, S. Orlic et J.-P. Seifert. *Simple Photonic Emission Analysis of AES.* In Cryptographic Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science, 2012. (Cité en page 13.)

[Schmidt 2007] Jörn-Marc Schmidt et Michael Hutter. *Optical and EM Fault-Attacks on CRT-based RSA : Concrete Results.* In Proceedings of the 15th Austrian Workhop on Microelectronics - Austrochip, Graz, Austria, 2007. (Cité en page 34.)

[Sedcole 2006] Pete Sedcole et Peter YK Cheung. *Within-die delay variability in 90nm FPGAs and beyond.* In Field Programmable Technology (FPT). IEEE, 2006. (Cité en page 53.)

[Selmane 2008] N. Selmane, S. Guilley et J.-L. Danger. *Practical Setup Time Violation Attacks on AES.* In European Dependable Computing Conference (EDCC). IEEE Computer Society, 2008. (Cité en page 13.)

[Selmane 2011] N. Selmane, S. Bhasin, S. Guilley et J.L. Danger. *Security evaluation of application-specific integrated circuits and field programmable gate arrays against setup time violation attacks.* Information Security, IET, 2011. (Cité en pages 19, 50 et 88.)

[Shang 2002] Li Shang, Alireza S Kaviani et Kusuma Bathala. *Dynamic Power Consumption in Virtex -II FPGA Family.* 2002. (Cité en page 104.)

[Skorobogatov 2003] S. Skorobogatov et R. Anderson. *Optical Fault Induction Attacks.* In Cryptographic Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science, 2003. (Cité en page 13.)

[Standaert 2009] François-Xavier Standaert, Tal G Malkin et Moti Yung. *A unified framework for the analysis of side-channel key recovery attacks*. In Advances in Cryptology-EUROCRYPT. 2009. (Cité en page 110.)

[Stephenson 2009] JCDFR Stephenson, D Chen, R Fung et J Chromczak. *Understanding Metastability in FPGAs*. Altera Corporation white paper, 2009. (Cité en page 30.)

[Stinson 2005] D. Stinson. Cryptography : Theory And Practice. 2005. (Cité en page 10.)

[Tummeltshammer 2009] Peter Tummeltshammer et Andreas Steininger. *On the role of the power supply as an entry for common cause faults*. Design and Diagnostics of Electronic Circuits and Systems, 2009. (Cité en page 34.)

[Tunstall 2005] M. Tunstall et H. Choukri. *Round Reduction Using Faults*. In Fault Diagnosis and Tolerance in Cryptography (FDTC), 2005. (Cité en page 19.)

[Yanci 2009] Asier Goikoetxea Yanci, Stephen Pickles et Tughrul Arslan. *Characterization of a Voltage Glitch Attack Detector for Secure Devices*. Bio-inspired Learning and Intelligent Systems for Security, 2009. (Cité en page 34.)

[Yen 2000] Sung-Ming Yen et Marc Joye. *Checking before output may not be enough against fault-based cryptanalysis*. Computers, IEEE Transactions on, 2000. (Cité en page 18.)

[Zick 2013] Kenneth M Zick, Meeta Srivastav, Wei Zhang et Matthew French. *Sensing nanosecond-scale voltage attacks and natural transients in FPGAs*. In Field programmable gate arrays (FPGA). ACM, 2013. (Cité en pages 45, 46 et 104.)

LABORATOIRE SAS - GARDANNE
ÉCOLE NATIONALE SUPÉRIEURE DES MINES DE SAINT-ÉTIENNE

NNT : 2014 EMSE 0757

Loïc ZUSSA

Étude des techniques d'injection de fautes par violation de contraintes temporelles permettant la cryptanalyse physique de circuits sécurisés

Spécialité :
Microélectronique

Mots clefs :
Attaque en faute, Glitchs de tension, Glitchs électromagnétiques, Glitchs d'horloge, Contraintes temporelles, FPGA

Résumé :

Si un algorithme cryptographique peut être mathématiquement sûr, son implémentation matérielle quant à elle est souvent la cible de nombreuses attaques. Cette thèse porte sur l'étude des mécanismes d'injection de fautes pouvant permettre une cryptanalyse physique des circuits sécurisés et sur la conception de contre-mesures matérielles pour empêcher ou détecter ces attaques.

Dans un premier temps une mise en pratique d'injection de fautes sur une implémentation matérielle de l'AES a été menée à l'aide d'attaques physiques non-invasives : variations statiques et dynamiques de la tension, de la fréquence, de la température et de l'environnement électromagnétique. La comparaison des fautes injectées nous a permis de conclure que ces différentes attaques partagent un mécanisme d'injection identique : la violation de contraintes temporelles.

La conception et l'implémentation d'un voltmètre intégré nous a permis d'observer les perturbations internes dues aux attaques par variations transitoires de la tension. Ces observations ont permis une meilleure compréhension du mécanisme

d'injection de fautes associé et une amélioration de la précision temporelle de ces injections.

Ensuite, un détecteur a été implémenté et son efficacité face à des attaques électromagnétiques a été étudiée. Du fait de la localité spatiale de ces attaques, la zone effectivement protégée par le détecteur est limitée. Une implémentation de plusieurs détecteurs a été suggérée pour étendre la zone de protection.

Enfin, un nouveau chemin d'attaque exploitant la sensibilité du détecteur a été proposé et validé expérimentalement. Cette attaque utilise le couplage électrique existant entre l'AES et le détecteur. A cause de ce couplage électrique la sensibilité de l'alarme varie en fonction des données traitées par l'AES.

LABORATOIRE SAS - GARDANNE
ÉCOLE NATIONALE SUPÉRIEURE DES MINES DE SAINT-ÉTIENNE

NNT : 2014 EMSE 0757

Loïc ZUSSA

Study of fault injections means based on timing constraints violation for physical cryptanalysis of secure circuits

Speciality :
Microelectronics

Key words :
Fault attack, Power supply glitches, Electromagnetic glitches, Clock glitches, Timing constraints, FPGA

Abstract :

Even if a cryptographic algortihm could be mathematically secure, its physical implementation could be targeted by several attacks. This thesis focus on time-based fault injection mechanisms used for physical cryptanalysis of secure circuits.

First, practical fault injections have been performed on a hardware AES implementation using non-invasive attacks : static and dynamic variations of the power supply voltage, frequency, temperature and electromagnetic environement. Then a comparison of these obtained faults led us to conclude that these different injection means share a common injection mecanism : timing constraints violations.

An on-chip voltmeter has been designed and implemented to observe internal disturbences due to voltage glitchs. These observations led to a better understanding of the fault injection mecanism and to a better temporal accuracy.

Then, a contermeasure has been designed and its effectiveness against electromagnetic attacks has been studied. Because of the electromagnetic pulses local effects, the aera effectively protected by the countermeasure is limited. The imple-

mentation of several countermeasures has been considered in order to extend the protected aera.

Finally, a new attack path using the countermeasure detection threshold variations has been proposed and experimentaly validated. This attack exploit the electrical coupling between the AES and the coutnermeasure. Because of this coupling the countermeasure sensitivity variations are related to data handled by the AES.